НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп‘ютерних систем

«До захисту допущено»

Завідувач кафедри

_____ Тарасенко В.П.
(підпис) (ініціали, прізвище)

“___” червня 2019 р.

Дипломний проект

на здобуття ступеня бакалавра

з напряму підготовки **6.050102 «Комп'ютерна інженерія»**

на тему: Комп’ютерна система ідентифікації користувача на основі голосового сигналу

Виконав: студент IV курсу, групи КВ-53
(шифр групи)

Тодорів Андрій Дмитрович _____ _____
(прізвище, ім’я, по батькові) (підпис)

Керівник проф. каф. СПіСКС, д. т. н., проф Терейковський І.А. _____
_____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант з нормоконтролю, доц.каф.СПСКС, к.т.н. доц. Клятченко Я.М.
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент_____
_____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цьому дипломному проекті немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2019 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Напрям підготовки (програма професійного спрямування) – 6.050102 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ В.П. Тарасенко
«___»_____ 2019 р.

ЗАВДАННЯ

на дипломний проект студенту

Тодоріву Андрію Дмитровичу

1. Тема проекту «Комп'ютерна система ідентифікації користувача на основі голосового сигналу», керівник роботи проф. каф. СПіСКС, д. т. н., проф. Терейковський І.А, затверджені наказом по університету від «22» _травня_ 2019 р. №1330-С

2. Термін подання студентом роботи «15» червня 2019 р.

3. Вихідні дані до роботи: див. Технічне завдання

4. Зміст пояснювальної записки:
   - аналіз існуючих рішень та обгрунтування теми дипломного проекту;
   - впровадження нейронної мережі для вирішення проблеми розпізнавання голосу;
   - опис розроблених алгоритмів;
   - аналіз розробленої системи.

5. Перелік обов'язкового ілюстративного матеріалу:
   - узагальнена схема швидкого перетворення Фур'є. Схема структурна.
   - алгоритм тренування нейронної мережі. Схема структурна.
   - алгоритм розпізнавання голосового сигналу. Схема структурна.

– архітектура системи розпізнавання голосового сигналу. Схема структурна.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Н.контроль	Клятченко Я.М., к.т.н., доцент		

7. Дата видачі завдання «31» жовтня 2018 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Вивчення літератури за тематикою роботи та збір даних	31.10.2018	
2.	Проведення порівняльного аналізу математичних методів розпізнавання голосового сигналу	09.12.2018	
3.	Підготовка матеріалів першого розділу дипломної роботи	30.12.2018	
4.	Розробка математичної моделі дискретизації вхідного сигналу	16.01.2019	
5.	Підготовка матеріалів другого розділу дипломної роботи	10.02.2018	
6.	Розробка нейромережевої технології	20.02.2019	
7.	Підготовка матеріалів третього розділу дипломної роботи	10.03.2019	
8.	Підготовка матеріалів четвертого розділу дипломної роботи	11.04.2019	
9.	Підготовка графічної частини дипломної роботи	19.05.2019	
10.	Оформлення дипломної роботи	26.05.2019	

Студент А.Д. Тодорів

Керівник роботи І.А. Терейковський

АНОТАЦІЯ

Дана кваліфікаційна робота досліджує та пропонує розв’язання проблеми розпізнавання голосу.

В процесі виконання дипломного проекту був розроблений програмний комплекс розпізнавання голосових сигналів в режимі реального часу за допомогою рекурентних нейронних мереж.

Під час розробки були сформовані вимоги до програмного комплексу, виконано порівняльний аналіз існуючих рішень, досліджені технологічні шляхи розв’язання задачі, та обрані програмні алгоритми, що задовольняють поставленим вимогам: низький рівень помилок розпізнавання, що досягається шляхом нівелювання зовнішнього впливу, та аналітична швидкодія програми.

Розроблена комп’ютерна система вирішує проблему розпізнавання голосу, за допомогою рекурентних нейронних мереж, що дозволяє використовувати її для спрощення роботи з електронним обчислювальним пристроєм.

Ключові слова:

РОЗПІЗНАВАННЯ МОВЛЕННЯ, НЕЙРОННА МЕРЕЖА, РЕКУРЕНТНА НЕЙРОННА МЕРМЕЖА

ABSTRACT

This qualification work explores and proposes the solution of the problem of recognition of a human voice.

In the process of implementing the diploma project, a program of recognition of voice signals in real time was developed using recurrent neural networks.

During the development, the requirements for the software complex were formed, a comparative analysis of existing solutions was performed, technological solutions for the problem were investigated, and selected software algorithms that meet the requirements: low level of recognition errors achieved by leveling external influences, and analytical program performance.

The developed computer system solves the problem of voice recognition, using recurrent neural networks, which allows it to be used to simplify the operation of the electronic computing device.

Keywords:

SPEECH RECOGNITION, NEURAL NETWORK, RECURRENT NEURAL MEMORY

№ п/п	Формат	Позначення	Найменування	Кількість листів	Примітка
			Документація загальна		
			Новорозроблена		
1		ІАЛЦ.045490.002 ТЗ	Комп'ютерна система ідентифікації користувача на основі голосового сигналу. Технічне завдання	4	
2		ІАЛЦ.045490.003 ВП	Комп'ютерна система ідентифікації користувача на основі голосового сигналу .	1	
			Відомість технічного проекту		
3	А4	ІАЛЦ.045490.004 ПЗ	Комп'ютерна система ідентифікації користувача на основі голосового сигналу .	61	
			Пояснювальна записка		
4	А1	ІАЛЦ.045490.005 Д1	Узагальнена схема швидкого перетворення Фур'є. Схема структурна.	1	
5	А1	ІАЛЦ.045490.006 Д1	Алгоритм тренування нейронної мережі. Схема структурна.	1	
6	А1	ІАЛЦ.045490.007 Д1	Алгоритм розпізнавання голосового сигналу. Схема структурна.	1	
7	А1	ІАЛЦ.045490.008 Д1	Архітектура системи розпізнавання голосового сигналу. Схема структурна.	1	

		CD-ROM			Матеріали бакалаврського проекту	1	
					ІАЛЦ.045490.001 ОА		
Зм.	Арк	№ докум	Підпис	Дата			
Розроб.		Тодорів А.Д.			*Комп'ютерна система ідентифікації користувача на основі голосового сигналу* ***Опис альбому***	***Літ.***	***Арк.*** 1 / ***Аркушів*** 1
Перевір.		Терейковський І.А					
Н. контр.		Клятченко .М.				КПІ ім. Ігоря Сікорського, ФПМ, КВ-53	
Затв.		Тарасенко В.П.					

ЗМІСТ

1. НАЙМЕНУВАННЯ І ОБЛАСТЬ ЗАСТОСУВАННЯ

Найменування розробки – «Комп’ютерна система ідентифікації користувача на основі голосового сигналу». Область застосування: голосове управління електронною обчислювальною машиною.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є актуальність дослідження існуючих програмних рішень для подальшої модифікації, з ціллю зниження рівня помилкового розпізнавання.

3. ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ

Підставою для розроблення є завдання на дипломне проектування, затверджене кафедрою системного програмування та спеціалізованих комп’ютерних систем Національного технічного університету України «Київський Політехнічний Інститут ім. Ігоря Сікорського».

4. ДЖЕРЕЛА РОБОТИ

Джерелом інформації в ході розробки були: електронні ресурси, періодичні видання, та технічна література.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється:

Комп’ютерна система повинна забезпечувати такі основні функції:

- Можливість взаємодії з зовнішнім пристроєм – мікрофоном;
- Можливість аналізу голосового сигналу в режимі реального часу;
- Наявність помилок розпізнавання – не більше 5%;
- Мати зручний інтерфейс користувача.

5.2. Вимоги до апаратного забезпечення

Вимоги до електронного обчислювального пристрою, на якому буде використовуватись розроблена програмно-апаратна система:

- процесор з тактовою частотою 2 ГГц або вище,
- оперативна пам’ять обсягом 4 Гб або більше

5.3. Вимоги до програмного забезпечення:

- Операційна система Windows 8.1;
- Python 3.7.3:
- Мати встановлену систему управління пакетами pip, та потокову бібліотеку pyAudio;

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів роботи та питань, які мають бути розроблені відповідно до завдання	Термін виконання
1.	Видача завдання на дипломне проектування	15.10.2019
2.	Розробка технічного завдання	16.04.2019
3.	Аналіз існуючих рішень	17.04.2019
4.	Вибір середовища розробки	25.04.2019
5.	Розробка методу використання нейронних мереж	26.04.2019
6.	Розробка методу визначення вхідних та вихідних параметрів НМ, призначених для розпізнавання голосових сигналів	03.05.2019
7.	Розробка програмного продукту	10.05.2019
8.	Відлагодження програмного продукту	20.05.2019
9.	Підготовка пояснювальної записки	22.05.2019

10.	Оформлення матеріалів проекту	25.05.2019
11.	Попередній огляд матеріалів диплому на кафедрі	31.05.2019

№ п/п	Фор	Позначення	Найменування	Кількість листів	Примітка
			Документація загальна		
			Новорозроблена		
1	А4	ІАЛЦ.045490.004 ПЗ	Комп'ютерна система ідентифікації користувача на основі голосового сигналу.	61	
			Пояснювальна записка		
			проекту		
2	А1	ІАЛЦ.045490.005 Д1	Узагальнена схема швидкого перетворення Фур'є. Схема структурна.	1	
3	А1	ІАЛЦ.045490.006 Д1	Алгоритм тренування нейронної мережі. Схема структурна.	1	
4	А1	ІАЛЦ.045490.007 Д1	Алгоритм розпізнавання голосового сигналу. Схема структурна.	1	
5	А1	ІАЛЦ.045490.008 Д1	Архітектура системи розпізнавання голосового сигналу. Схема структурна.	1	
		CD-ROM	Матеріали бакалаврського проекту	1	

ІАЛЦ.045490.003 ВП

Зм.	Арк	№ докум	Підпис	Дата				
Розроб.		Тодорів А.Д.			*Комп'ютерна система ідентифікації користувача на основі голосового сигналу*	*Літ.*	*Арк.*	*Аркушів*
Перевір.		Терейковський І.А.					1	1
					Відомість проекту	КПІ ім. Ігоря Сікорського, ФПМ КВ-53		
Н. контр.		лятченко .М.						
Затв.		Тарасенко В.П.						

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

ЕОМ – електронна обчислювальна машина

AI – штучний інтелект (artificial intelligence)

WER – рівень помилкового розпізнавання (word error rate)

DNN – нейронні мережі глибокого навчання (deep neural networks)

DFT – дискретне перетворення Фур'є (*discrete Fourier transform)*

HMM *– приховані моделі Маркова (hided Markov models)*

CTC *– алгоритм тимчасової класифікації (connectionists temporal classification)*

HM *– нейронні мережі*

PHM *– рекурентні нейронні мережі*

LSTM *– довга короткочасна пам'ять (long short term memory)*

GRU *– вентильний рекурентний вузол (gated recurrent units)*

SFTF *– віконне перетворення Фур'є (*Short-time Fourier transform)

FFT – *швидке перетворення Фур'є (fast Fourier transform)*

RMS *– середнє квадратичне (root mean square)*

MFCC – Мел-кепстральні частотні коефіцієнти (mel-frequency cepstral coefficients)

API – прикладний програмний інтерфейс (applied program interface)

Вступ

В даний час комп'ютерні технології використовуються в багатьох сферах людської діяльності, як зручний і багатофункціональний інструмент для широкого кола завдань. Проте користувачі комп'ютерів змушені застосовувати методи взаємодії, які погано адаптуються до можливостей людського спілкування і обмежують здатність людини до обміну інформацією. Основною метою вдосконалення та розвитку інтерфейсу «людина-комп'ютер» є організація обміну інформацією з комп'ютером таким чином, щоб:

- Зменшити часові затрати на розробку програмного та апаратного забезпечення;
- Зменшити рівень помилкового розпізнавання при передачі інформації;
- Зробити роботу з КС ефективною для людей з обмеженими можливостями;
- Зменшити стомлюваність від використання ЕОМ, збільшити суб’єктивне задоволення від користування КС.

Для досягнення даних цілей необхідно використовувати засоби взаємодії, які більш повно використовують комунікативні можливості людей. Людина наділена великою кількістю можливостей сприймати та передавати інформацію: зір, слух (у тому числі мова), жести і рухи, міміка, дотик та інші. При взаємодії людини і комп'ютера існують два інформаційні потоки:

- команди керування та дані, що передаються на комп'ютер для обробки;

- обчислювальні результати та інша інформація, що надається комп'ютером користувачу.

В даний час широко поширений людинно-машинний інтерфейс використовує зір як основний канал для надання інформації користувачеві, відображаючи дані у вигляді символів на екрані комп'ютера. Сприйняття інформації природними для людини шляхами (розпізнавання мови, жестів, міміки тощо) практично неможливе для сучасних прикладних інтерфейсів.

Людина є важливим джерелом інформації при спілкуванні між людьми. Вираз обличчя, артикуляція під час розмови, рух голови є зручним, природним і, що важливо, бездоганним способом передачі інформації. Нездатність комп'ютера, з одного боку, сприймати, а з іншого боку відтворити такий природний спосіб передачі людської інформації для спілкування, ускладнює передачу і сприйняття інформації при роботі з комп'ютером.

Для забезпечення ефективного усного діалогу між користувачем і комп'ютером потрібні стабільні системи розпізнавання мови. Основною метою дипломного проекту є розробка системи розпізнавання голосових сигналів для подальшого керування ЕОМ.

АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБГРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ

1.1. Загальний опис проблеми ідентифікації користувачів за голосовим сигналом

Проблема ідентифікації голосового сигналу набула поширення у другій половині XX століття, ще в якості задачі розпізнавання мовлення. Перші практичні напрацювання в даній галузі були створені вже в кінці 80-х років на основі НММ.

Розпізнавання мовлення є однією з найбажаніших задач поставлених ЕОМ, що пов'язана з аналізом звуку, та мала різні шляхи розвитку: від цивільного та громадського, до приватного та військового застосування, це тема завжди була затребуваною для дослідження.

Поки AI помічник на телефонах аналізує наші побажання, військові бази даних збирають і зберігають інформацію про потенційних вбивць, торговців зброєю та наркотиками в пасивному режимі, що допомагає запобігати злочинам з мінімальними витратами часу.

Слід зазначити, що можливість побудови алгоритмів такого типу з'явилася не так давно, через апаратне забезпечення, здатне обробляти достатній обсяг даних, у необхідні терміни.

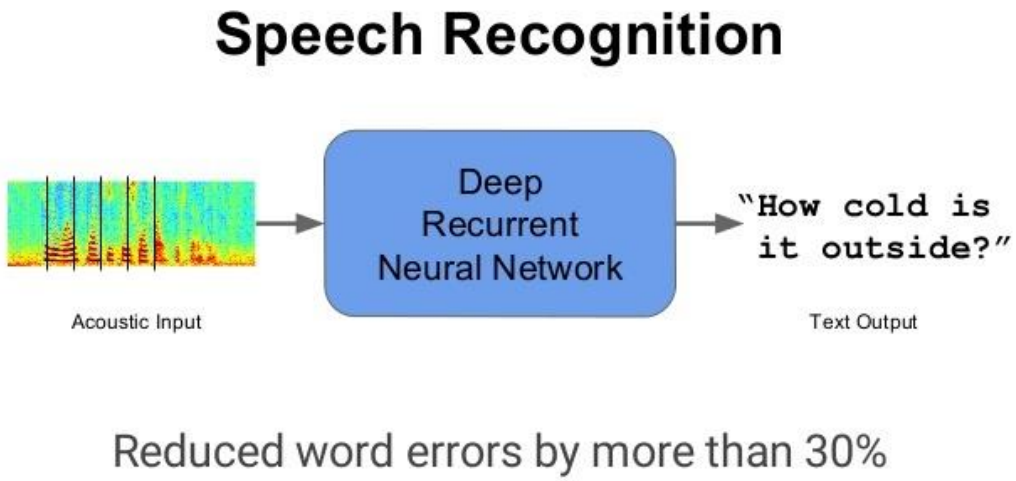
На вході ми маємо аналоговий акустичний аудіо сигнал, який необхідно перетворити в цифровий тип даних, що здатний бути проаналізованим рекурентною НМ, для подальшої інтерпретації вхідного сигналу у текстовому форматі, що сумісний з користувачем.

Таким чином, проблема буде класифікуватися за кількістю слів, що здатні бути проаналізованими. Їх кількість формує набір класів, що будуть застосовані у вирішенні проблеми. Ця проблема з'явилася 50 років тому, коли вчені створили систему розпізнавання мовлення “Watermelon”, здатну зрозуміти одне слово – “Watermelon”, що стало можливим завдяки спеціальному розміщенню голосних у цьому слові. У такому випадку програма мала лише один клас і два випадки, що створювало можливість її побудови на повністю аналоговій базі. Тож, збільшення кількості класів, веде

до потреби підвищення потужності апаратного забезпечення, та збільшення часових затрат, для успішного аналізу.

Рисунок 1.1 – Схема розпізнавання голосового сигналу за допомогою РНМ

В кінці 90-х років, алгоритми вже мали можливість розуміти до 100 слів. Але задача ставала непридатною до розв'язання з кількістю класів більшу за



1000, або у випадках, коли слова були сказані без пауз.

Головною метрикою в питаннях якості розпізнавання є WER. WER - це відсоток помилок розпізнавання. Як видно з рисунка 1.2, так звану «мову читання» було легко впізнавати, на відміну від розмовної мови, що створювало проблеми для військових досліджень, тому що людська мова в поєднанні з зовнішнім шумом, пропускається через мікрофон стільникового телефона, канали передачі якого використовують смуги вузької пропускної здатності, що практично унеможливлює подальший аналіз.

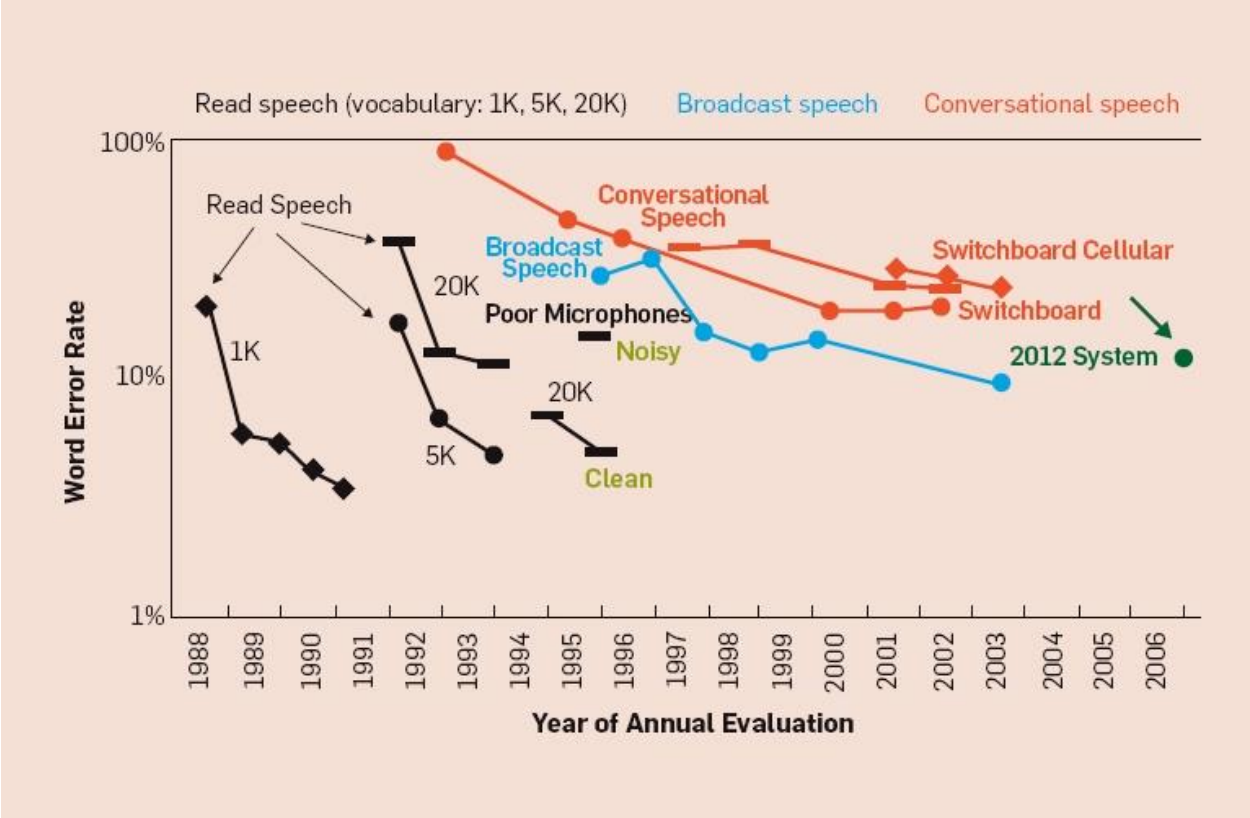


Рисунок 1.2 – Розвиток якості систем розпізнавання

Фактори, що впливають на WER:

- Розмір словника;
- Наявність пауз між словами;
- Якість запису;
- Наявність зовнішнього шуму;
- Особливості мови.

З 1999 до 2010 якість розпізнавання була сталою.

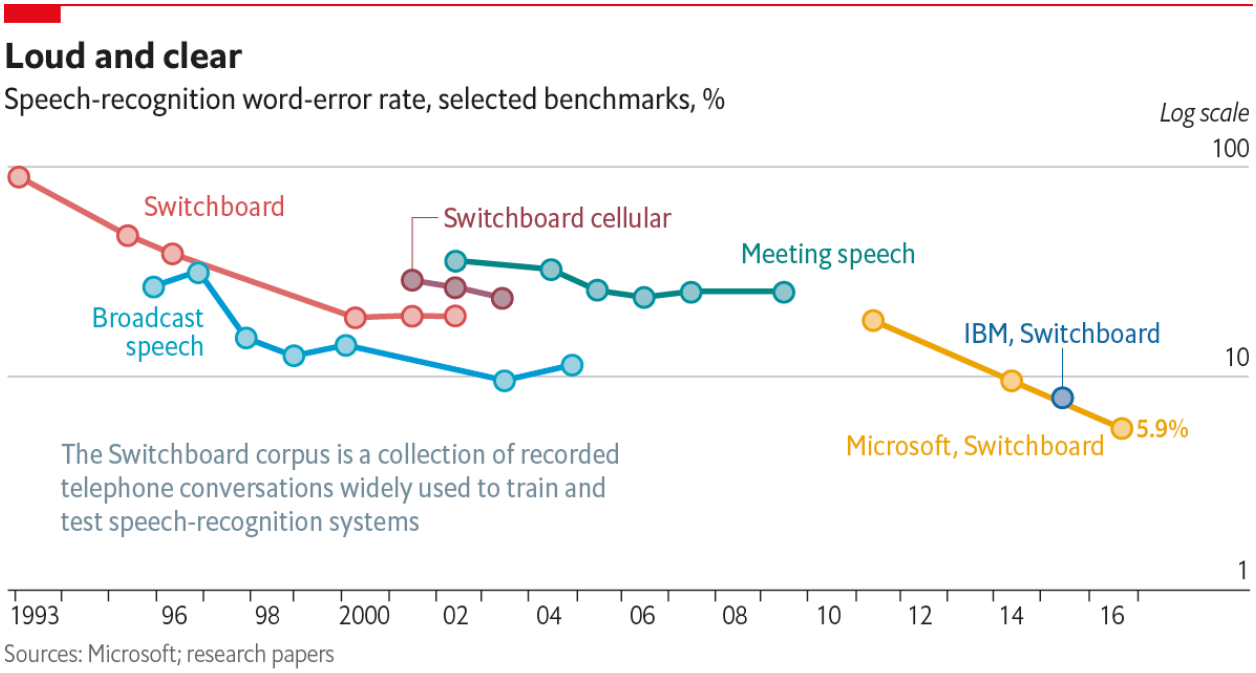


Рисунок 1.3 – Розвиток якості розпізнавання систем «чистої» мови

Швидкість розпізнавання зросла через появу нових апаратних засобів, що збільшило розміри словників, які можливо проаналізувати. Але якість н зросла. Також, розроблені алгоритми вимагали часових затрат на тренування НМ, щоб задовольняти WER. З рис. 1.4, видно недоліки старих технологій в порівнянні з DNN.

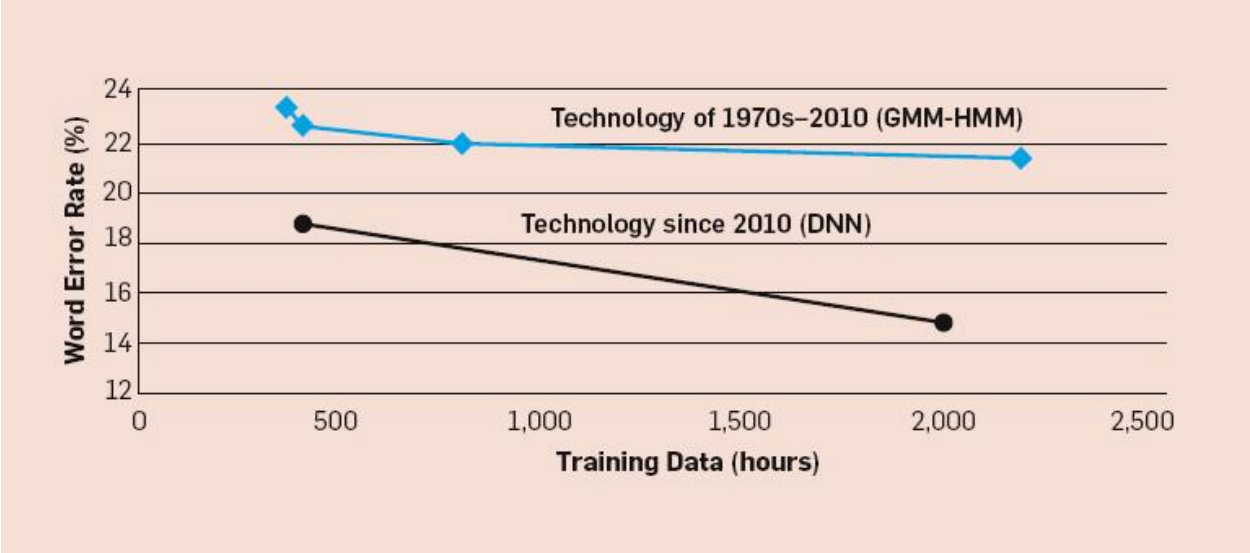


Рисунок 1.4 – Залежність WER від різних типів технологій

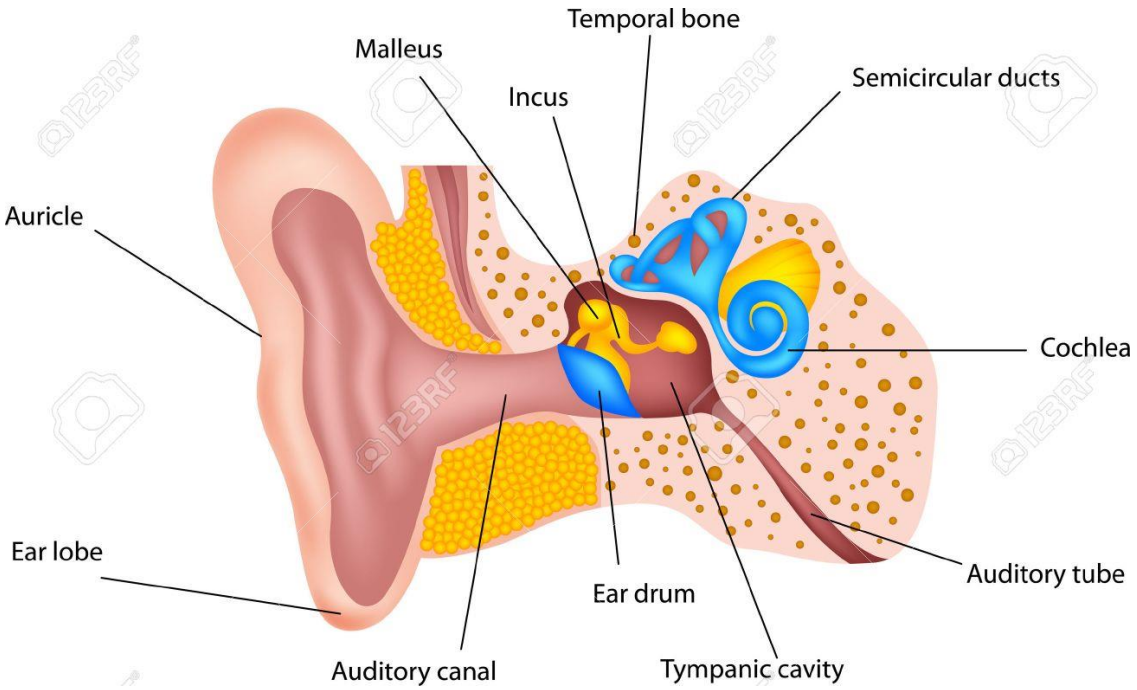
1.2. Способи розпізнавання голосового сигналу в комп'ютерних

системах

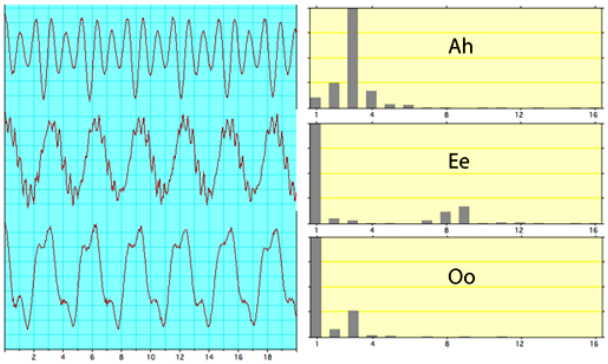
Звук - це фізична вібрація, хвиля, яку треба захопити для подальшого аналізу. Аналіз може бути виконаний після перетворення, що дозволяє розділити сигнал на різні частотні діапазони.

Людське вухо здатне виконувати це перетворення за своєю природою. Різні частини внутрішнього вуха реагують на різні частотні діапазони, виконуючи нелінійне перетворення з вхідними звуками, тому людина наділена спеціальним приймачем для кожного діапазону частот, як видно з рис.1.5. Рис. 1.6 і 1.7 описують зв'язок між осциляторами, які виробляють різні частоти, та окремими тонами.

Рисунок 1.5 – Внутрішня структура вуха



Спираючись на перетворення Фур’є, кожен звук може бути представлений у вигляді комбінації синусоїдальних коливань різної частоти.



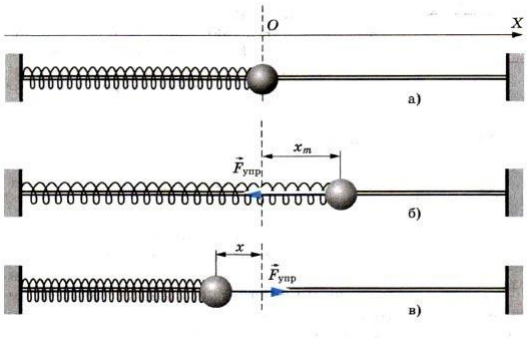
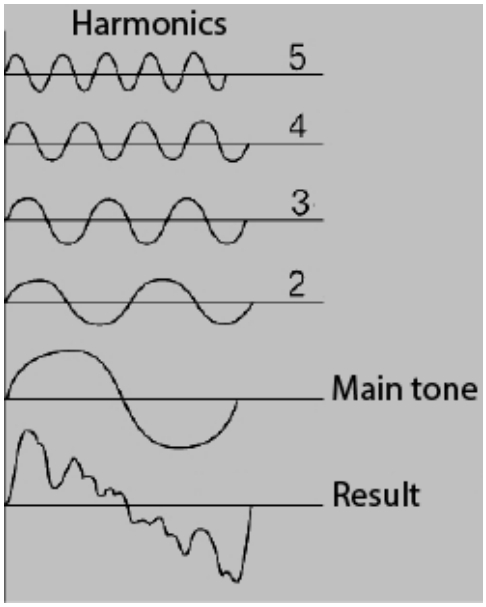


Рисунок 1.6 – Частотні залежності Picture 1.7 Механічні коливання

Кожна синусоїдна хвиля має різну амплітуду і частоту, що змушує звучати її в окремому тоні. Після комбінації хвилі будуть резонувати одна з одною, створюючи піки або ями на частотному діапазоні. Ця характеристика дозволяє сприймати інформацію шляхом відокремлення різних частотних діапазонів, тому модуль трансформації розробленої комп'ютерної системи повинен мати такі можливості:

- Аналіз частотного діапазону від 40 Гц до 600 Гц;
- Розкладання вхідного сигналу на окремі гармоніки.



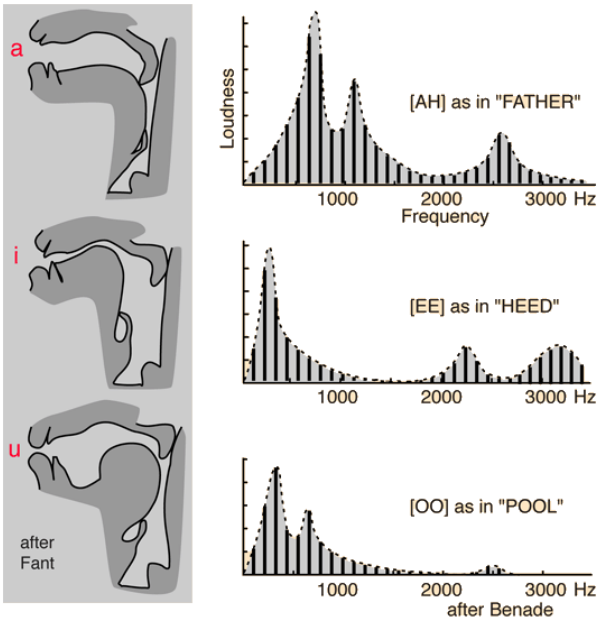
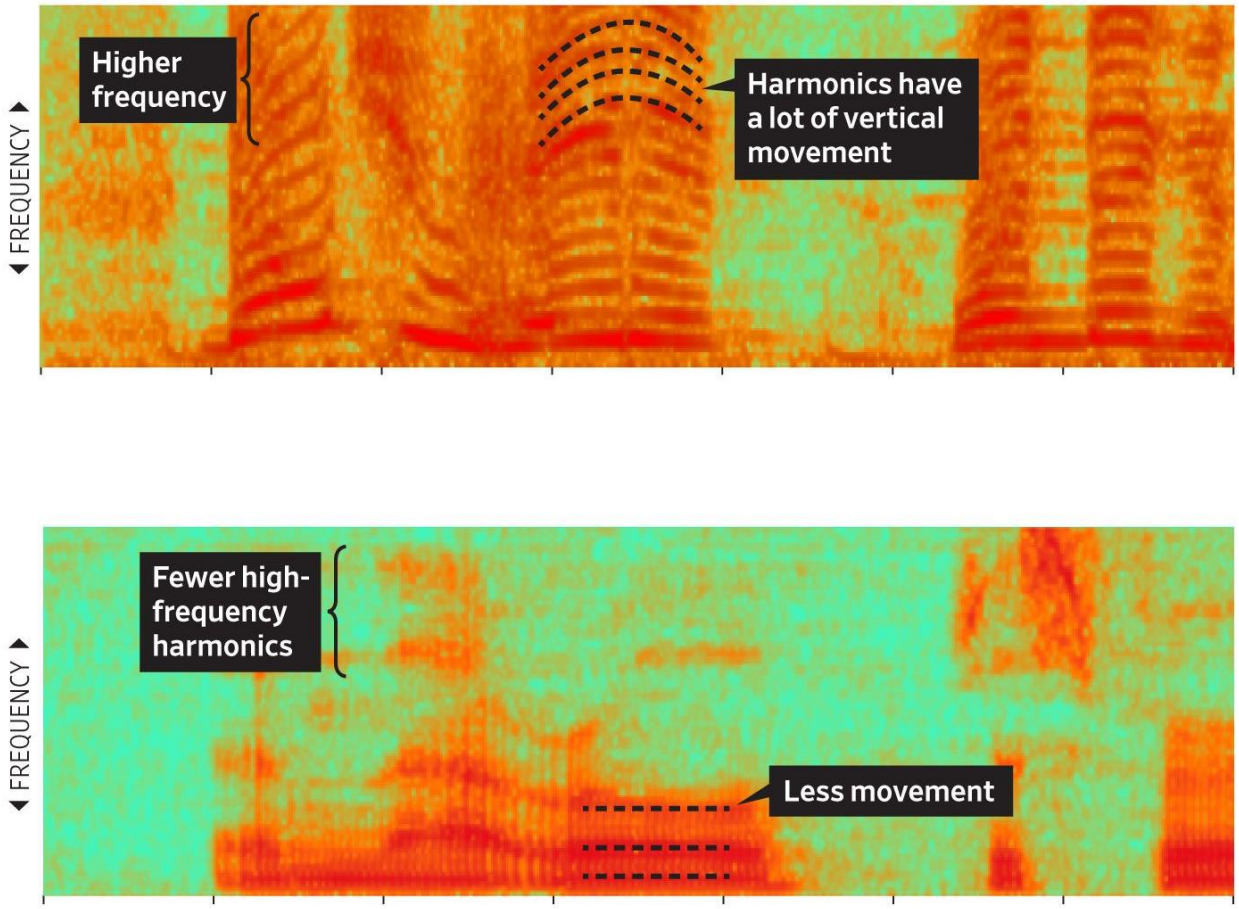


Рисунок 1.8 - Розкладання вхідного сигналу Рисунок 1.9 Частотний розподіл між звуками

Рисунок 1.8 зображує ситуацію, коли всі голоси повинні бути розпізнані по-різному, оскільки різні гармоніки кожного людського голосу впливають на наші рецептори різним шляхом, щоб бути впізнаваними. На рисунку 1.9 показано зв'язок між різними типами звуків та їх частотними результуючими, де амплітуда зображена на осі Y, а частота - на X.

Отже, розроблена комп'ютерна система вимагає створення функцій для поділу звуків на букви, або послідовності букв, готових до об'єднання в слова.

Рисунок 1.10 - Розподіл різних голосних звуків на частотному діапазоні



Як видно з рис. 1.10, голосні звуки легше піддаються розпізнаванню, тому перші механізми розпізнавання мови були побудовані на цьому принципі, аналізуючи послідовності голосних..

Рисунок 1.11 - Залежність між амплітудою частоти і часом

Наступним кроком є розробка динамічного алгоритму, здатного аналізувати частотний спектр у кожний момент часу. Як видно з рис. 1.11, залежність між часом і амплітудою сигналу показує нам відмінність сигналу. Отже, як людина здатна розібрати аналоговий сигнал, комп'ютерна система повинна мати аналогічний спосіб інтерпретації інформації, розділяючи сигнал на дискретні частини для виконання аналізу. Існує загальний конвеєр для систем розпізнавання голосу, описаний на рисунку 1.12:

1. Мовний сигнал, який надсилається на вхід, може описувати тільки амплітуду;
2. Короткочасні мовні сегменти формуються шляхом дискретизації аналогового сигналу – поділ неперервного звуку на окремі частини малої довжини (10 мс). Всі дискретизовані частини множаться на функцію, яка прибирає частоти, що містять вторинні гармоніки і шум;
3. Виконання DFT дає спектральну залежність між амплітудою і частотою для кожної аудіо-частини;
4. Складання спектральних залежностей дискретних частинок в динамічну картину забезпечує співвідношення між кожною амплітудою частоти і часом, що називається спектрограмою, що зображено на малюнку 1.11;
5. Вихідний сигнал необхідно відфільтрувати, щоб зробити його впізнаваним ЕОМ, відрізаючи частоти. З, приблизно, 500 Гц, залишаємо не більше 50 Гц, стискаючи його необхідним фільтром, описаним формулою. Цей етап називається фільтровим накопиченням, або Мел-трансформацією;
6. Мел-трансформація дозволяє алгоритму стискати вихідний сигнал, зменшуючи величину формуючих частот;
7. Класичний спосіб побудови алгоритму вимагає застосування MFCC, який стискає звук до 13 формуючих частот, за DCT, що шукає повторювані гармоніки і видаляє їх, щоб досягти результату в 13 частот.

Отримана інформація готова до аналізу. Але тоді у нас є два способи вирішення завдання, перетворення звуку в:

1. Фонеми;
2. Букви.

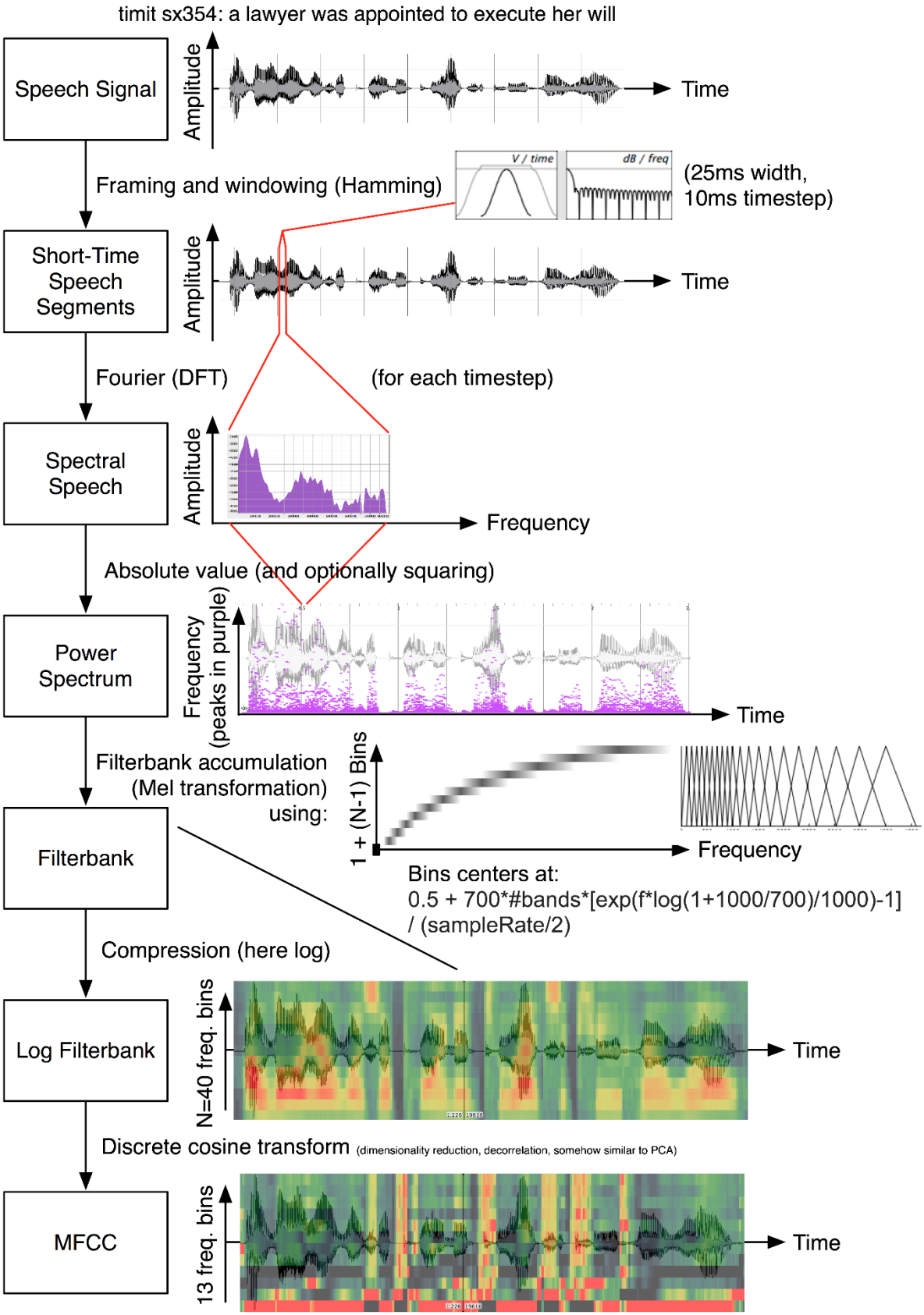


Рисунок 1.12 – Алгоритм транформації звуку

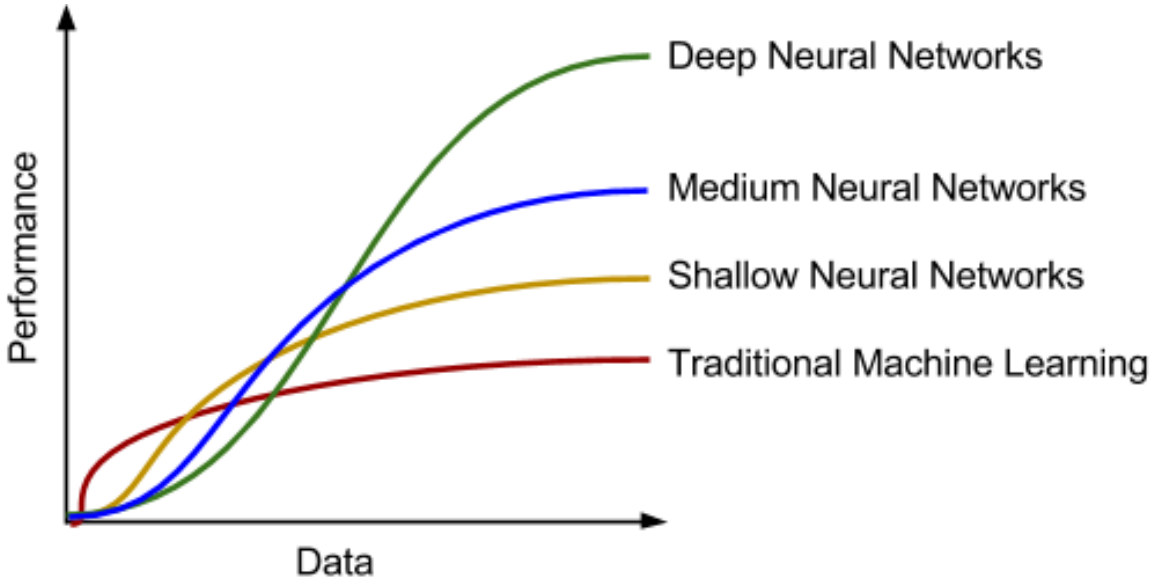


Рисунок 1.13 – Доцільність технологій в задачі розпізнавання мовлення

1. Перший метод передбачує розпізнавання фонем, для побудови потрібного слова. Процес конструкції сам по собі означає порівняння фонем з існуючими словниками, які називаються датасетами. Таким чином, цінність розпізнавання залежить від розміру набору даних, оскільки вона стає більшою, зі збільшенням слів, які готова визначити.

Одними з перших датасетів були:

- TIMIT Database (англ.) – 3 години, 1993;
- 199CMUDict (англ.) – 100 тис. слів, 1993.

Але, як видно з рис. 1.13, продуктивність повністю залежить від типу алгоритму, технологія дає нам можливість збирати більші словники, які можна швидше аналізувати, використовуючи нові технології.

2. Другий метод вимагає розпізнавання букв з аудіо сигналу.

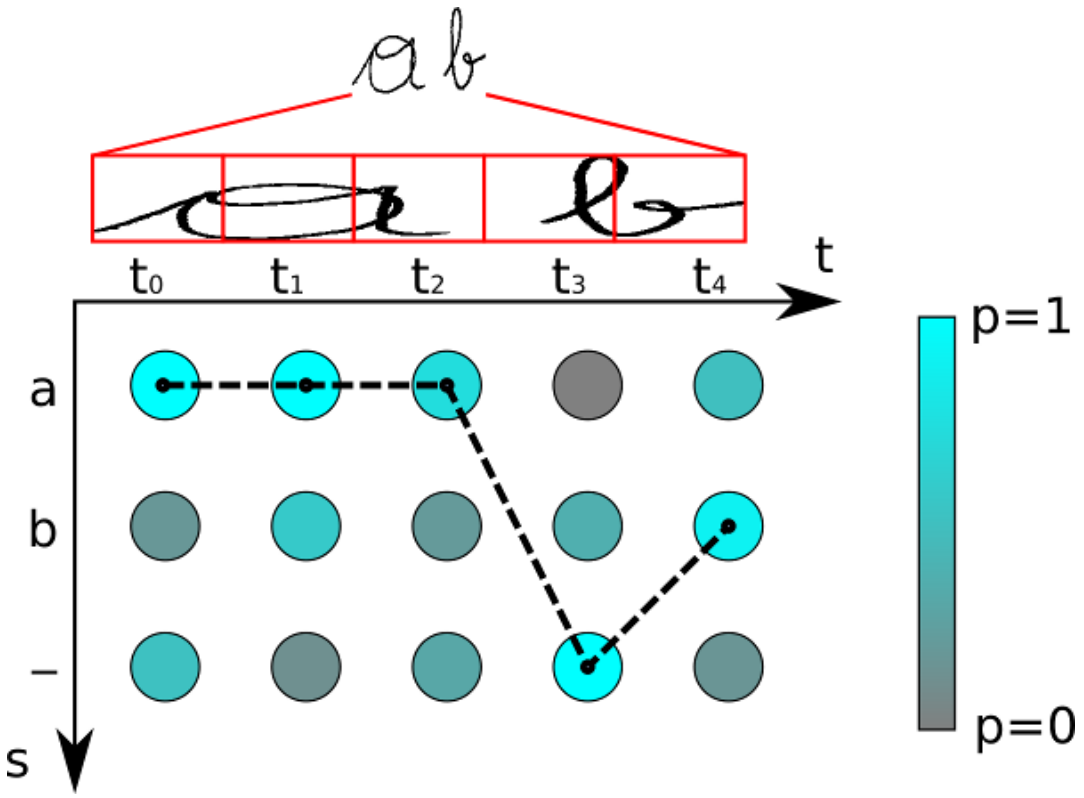
Цей метод збільшує розмір словника, що вимагає більшої апаратної потужності для обробки даних. Першими словниками без фонетичного аналізу були:

- WSJ (англ.) - 81 годин, 1993 ;

- Switchboard(англ.) - 240 годин, 1993;
- VoxForge (рос.) - 17 годин, 2009;
- **Fisher (англ.) - 2000 годин,2004;**
- **LibriSpeech (англ.) - 960 годин, 2015;**
- **Open_TTS (рос.) - більше 3000 годин, 2019.**

Перші три датасети складаються з фраз, розділених на слова. Останні три складаються з неподіленої розмовної мови, яка, в першу чергу, має бути обробленою у потрібній формі. Назва цього процесу - вирівнювання. Воно забезпечує перетворення аудіо сигналу на окремі букви. Існує декілька алгоритмів вирівнювання (alignment) - алгоритм прямого зворотного зв'язку (Forward-Backward) та алгоритм Вітербі. Їх зміст полягає у створенні ймовірнісної таблиці, що складається з різних букв, і їх відповідності до ймовірності появи в звуковому сигналі, як це зображено на рис. 1.14.

Рисунок 1.14 – Процес вирівнювання



Крім того, існує інший алгоритм, званий HMM. Але його ефективність нижче, ніж в алгоритму прямого зворотного зв’язку, тому що HMM не може працювати з послідовностями більше 4 звуків, букв, фонем. Схема роботи HMM наведена на малюнку 1.15

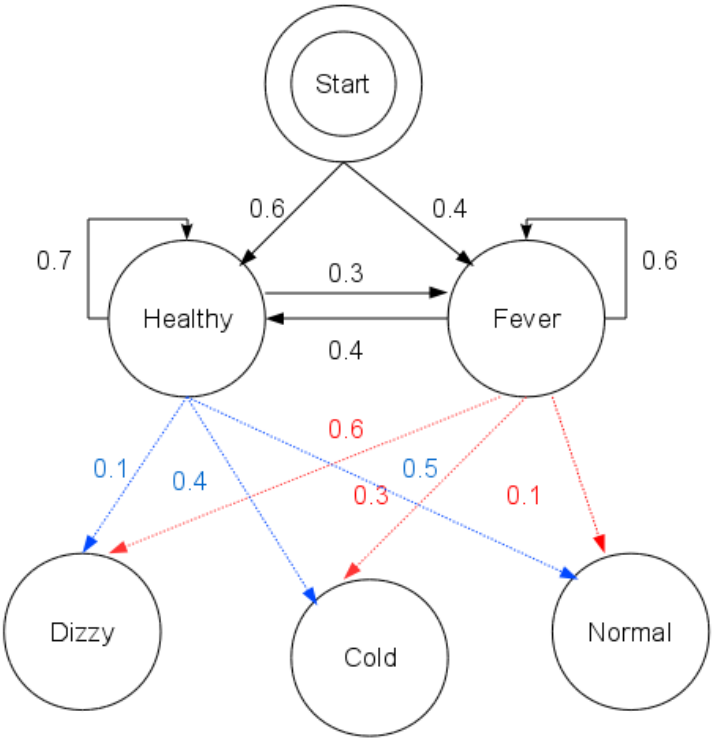


Рисунок 1.15 – Прототип алгоритму HMM

У зв'язку з недоліком HMM програма більш сумісна з алгоритмом Forward-Backward, що складає матриці ймовірності кожного звуку до кожної букви, що показано на рис. 1.16. Потім алгоритм знаходить найкоротший шлях через матрицю ймовірностей, мінімізуючи необхідні метрики.

Forward-Backward використовується в алгоритмі CTC, що описаний на малюнках 1.17 - 1.18, який порівнює вихідне слово, передбачене HM, з найбільш схожим словом, закріпленим у наборі даних. Потім алгоритм знаходить найменшу відстань редагування (edit distance) - цифру в клітинках таблиці, яка відповідає кількості перетворень, необхідних для завершення

вихідного слова. Щоразу, коли мережа робить помилку у визначенні відстані редагування, система отримує штраф, таким чином можливе її самонавчання.

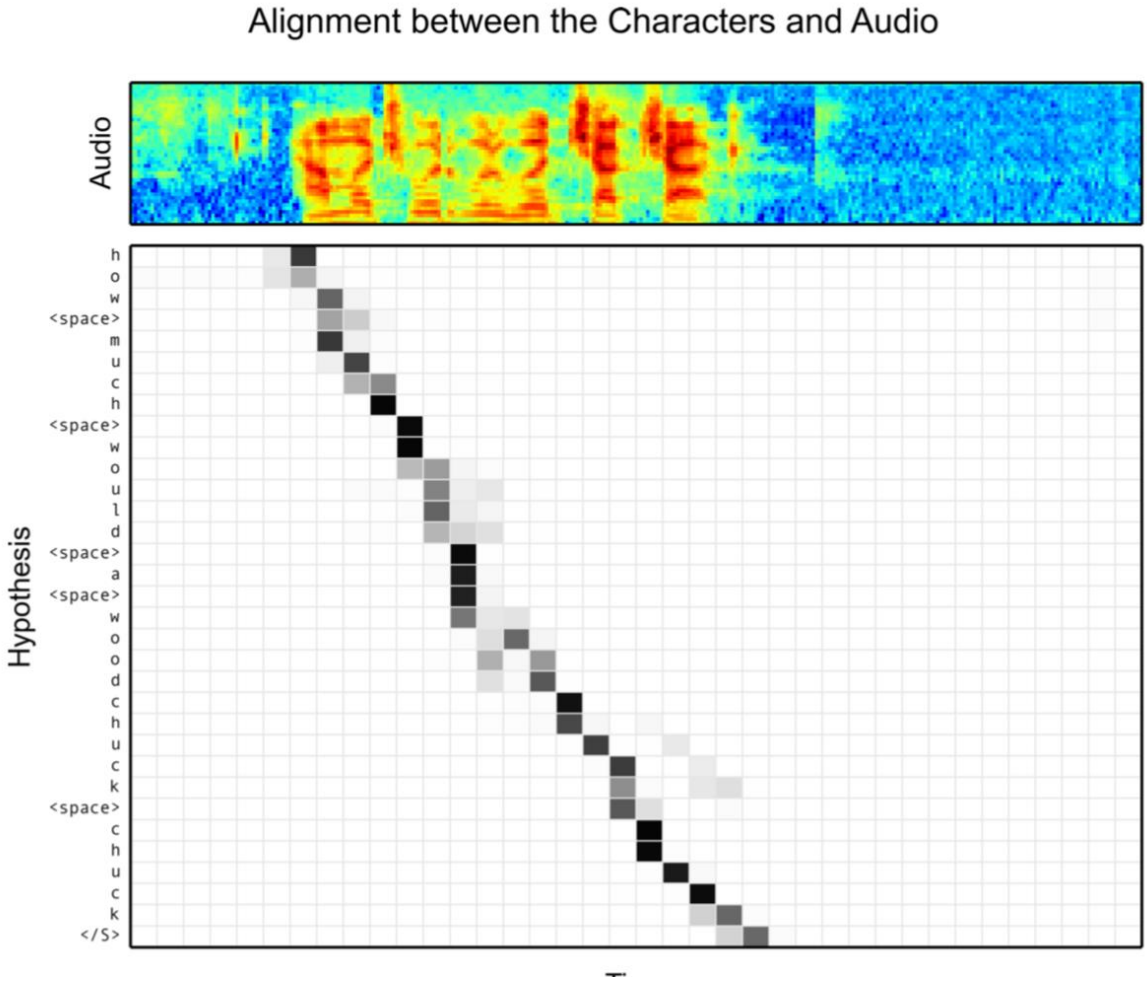


Рисунок 1.16 – Матриця ймовірностей

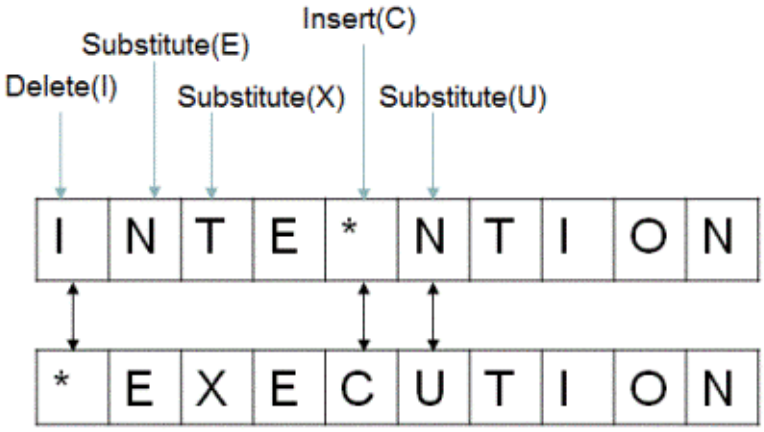


Рисунок 1.17 – Порівняння за допомогою алгоритму CTC

		E	L	E	P	H	A	N	T
	0	1	2	3	4	5	6	7	8
R	1	1	2	3	4	5	6	7	8
E	2	1	2	2	3	4	5	6	7
L	3	2	1	2	3	4	5	6	7
E	4	3	2	1	2	3	4	5	6
V	5	4	3	2	2	3	4	5	6
A	6	5	4	3	3	3	3	4	5
N	7	6	5	4	4	4	4	3	4
T	8	7	6	5	5	5	5	4	3

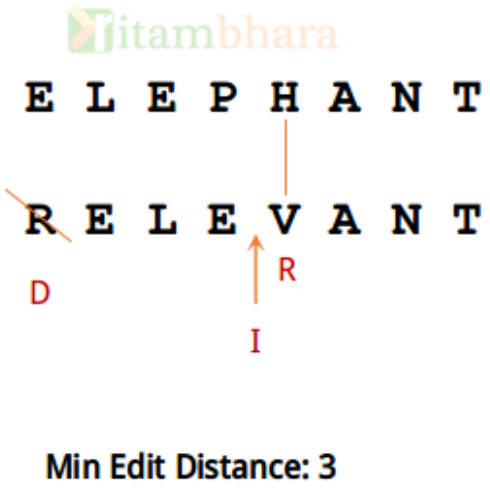


Рисунок 1.18 – Матриця відстаней редагування алгоритму CTC

1.3 Аналіз існуючих рішень розпізнавання голосового сигналу

Шуканий результат, та процес розпізнавання показаний на прикладі:

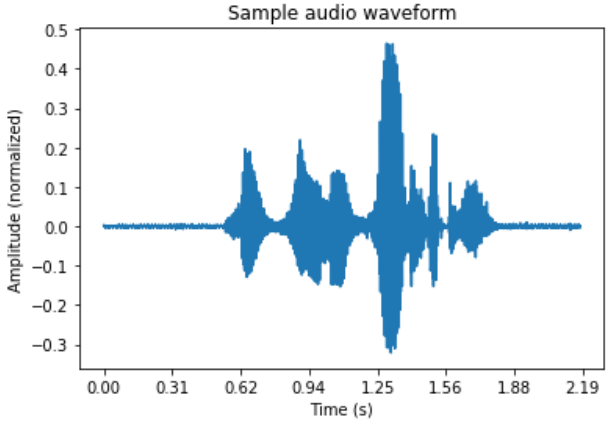


Рисунок 1.19 – Вхідний сигнал, залежність амплітуди від часу

На вхід ми подаємо аналоговий сигнал, що представляється як залежність між гучністю та часом. Дана форма сигналу перетворюється в послідовність цифрових проміжків, частини яких фіксуються на рис. 1.20. Ця здатність формувати динамічні послідовності дозволяє комп'ютерній системі порівнювати вхідні зразки з наведеними прикладами набору даних, як показано на рис. 1.21.

Датасет – це набір людського мовлення в режимі реального часу при різних умовах запису:

1. Розмовне, нерозмовне мовлення;
2. Читання;

3. Шумне спілкування.

Отже, потрібне застосування датасету. Існує два шляхи:

1. Побудова власної бази записів;
2. Використання існуючого датасету.

Перший шлях є важким через відсутність ресурсів, необхідних для його запису. Як видно з переліку, він вимагає різних людей, що вимовляють однакові фрази в різних умовах, а потім сортування інформації для того, щоб використовувати її за потрібним алгоритмом. Натомість вибір існуючих датасетів створює ще одне завдання – його фразова розмітка, розділення його на літери, які будують текст, як це видно на малюнку 1.20, що в свою чергу здатний бути порівняним з вхідним сигналом.

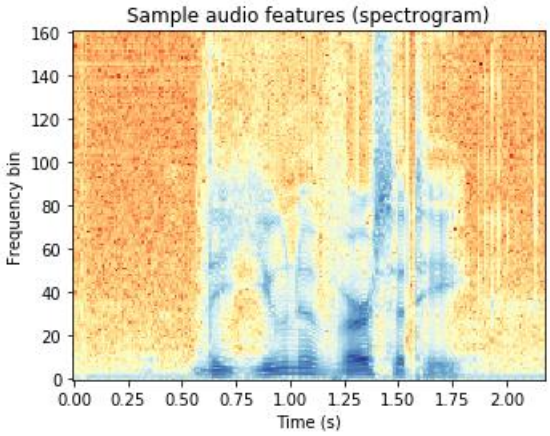


Рисунок 1.20 - Спектрограма

Коли набір даних побудований, мета полягає у визначенні ймовірності появи окремих літер, шляхом порівняння вхідного сигналу із зразками набору даних, те, що показано на рис. 1.21. Ця здатність реалізується шляхом введення НМ в систему. Тому необхідно дослідити технологію НМ.

Архітектура CTC для розпізнавання голосу заснована на основі алгоритму РНМ, який є типом НМ елементи якого формують орієнтований граф під час процесу аналізу. Ця здатність дозволяє обробляти такі динамічні послідовності, як голос, почерк.

Застосування CTC вимагає використання технологій LSTM та GRU:

1. LSTM мережа - це штучна НМ, яка складається з LSTM модулів, або є основою мережі. Модуль LSTM - це повторюваний мережевий модуль,

здатний запам'ятовувати значення на довгі або короткі часові інтервали.

2. GRU- вентильний рекурентний модуль - технологія на основі LSTM, яка має менше параметрів і не має вихідних даних, створений для забезпечення системи воріт в НМ.

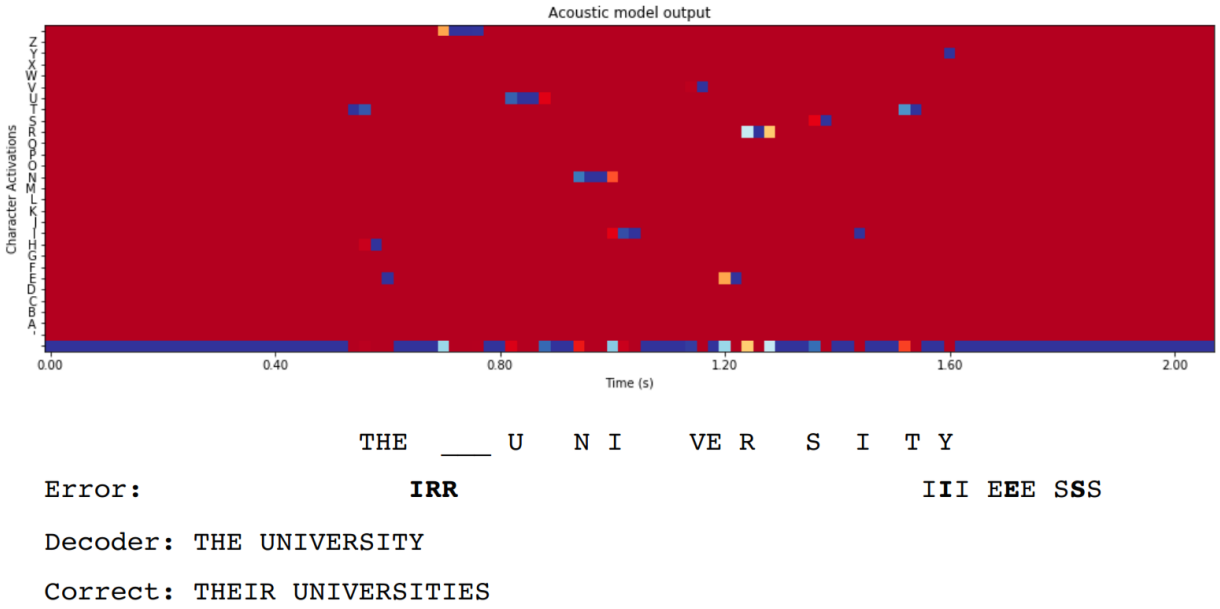


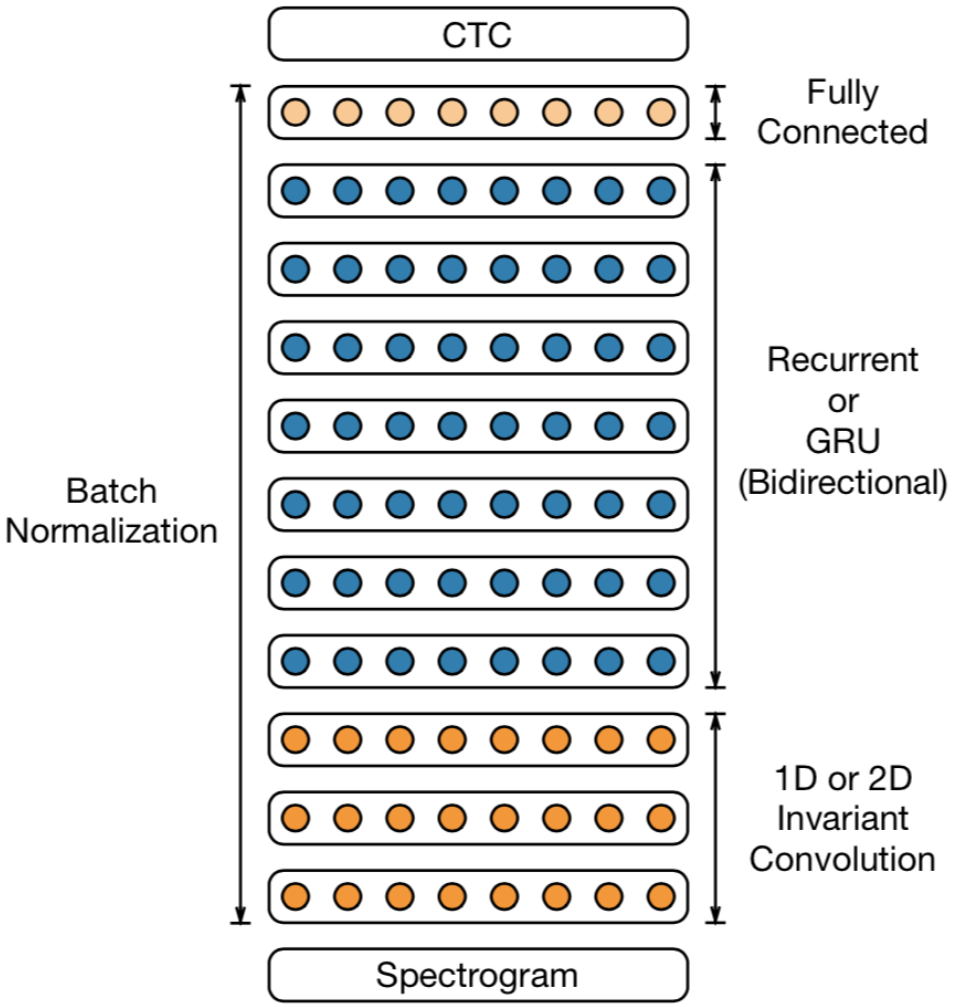
Рисунок 1.21 – Результат розпізнавання, зображений як залежність між часом та ймовірною літерою

Архітектура CTC, побудована за наступними технологіями знаходиться на рис. 1.22.

На вході є спектрограма, яка спочатку представлена як одно, або двовимірна структура, готова до зміни за допомогою згорткових шарів. Далі дані представлені в RNN або GRU - двонаправленою структурою, готовою до аналізу динамічних послідовностей. Процес нормалізації закінчується передачею з'єднання до повністю з'єднаного шару. Наступний метод

використання комбінації RNN, GRU, згорткових мереж добре показано на рис. 1.24, де одно або двовимірні моделі аналізуються за допомогою $h_t^{(1)} - h_t^{(3)}$- згорткових шарів , потім передаючи дані в двонаправлені шари $h_t^{(f)} - h_t^{(b)}$, і закінчуючи $h_t^{(5)}$- повністю зв'язаним нейронним шаром.

Рисунок 1.22 – Архітектура CTC



На виході отримана інформація про сигнал, що представлена як набір ймовірностей, і залишається тільки впровадження системи покарання і метрик, що зменшують WER. Після вибору бази алгоритму, система повинна бути здатна до самонавчання, тому вона потребує значних апаратних ресурсів, у вигляді процесорів з великою тактовою частотою роботи, а в ідеалі графічних прискорювачів, Також, в залежності від конфігурації системи, та словника кількість потрібного часу для тренування змінюється.

Architecture	Hidden Units	Train		Dev	
		Baseline	BatchNorm	Baseline	BatchNorm
1 RNN, 5 total	2400	10.55	11.99	13.55	14.40
3 RNN, 5 total	1880	9.55	8.29	11.61	10.56
5 RNN, 7 total	1510	8.59	7.61	10.77	9.78
7 RNN, 9 total	1280	8.76	7.68	10.83	9.52

Рисунок 1.23 – Аналіз архітектур PHM

Рис. 1.23 є результатом аналізу окремих параметрів архітектури. Найкращі результати продемонстровані за допомогою двовимірних згорток, що складаються з 7 рекурентних шарів, 1280 прихованих шарів, 68 млн. параметрів. Спираючись на це, інші архітектури повинні бути проаналізованими.

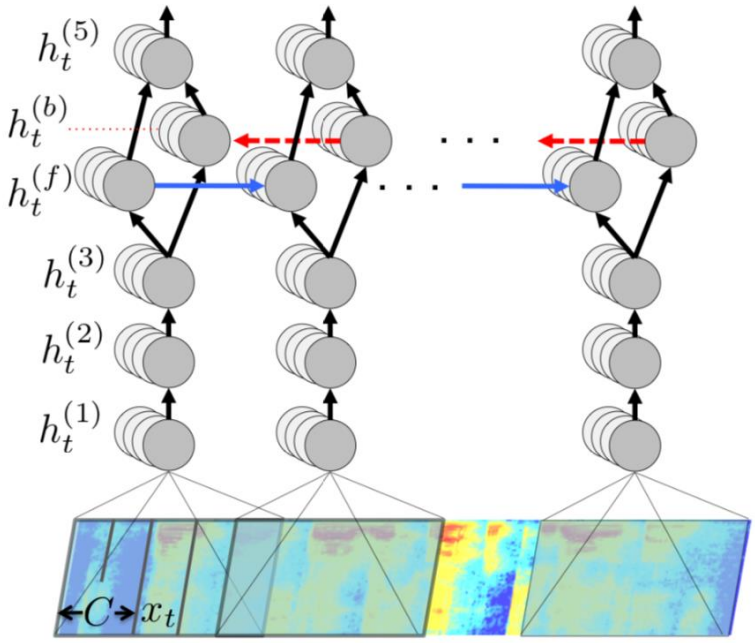


Рисунок 1.24 – Структура PHM

Dataset	Speech Type	Hours
WSJ	read	80
Switchboard	conversational	300
Fisher	conversational	2000
LibriSpeech	read	960
Baidu	read	5000
Baidu	mixed	3600
Total		11940

Fraction of Data	Hours	Regular Dev	Noisy Dev
1%	120	29.23	50.97
10%	1200	13.80	22.99
20%	2400	11.65	20.41
50%	6000	9.51	15.90
100%	12000	8.46	13.59

Рисунок 1.25 – Залежність між розміром датасету та якістю розпізнавання

Існує зв'язок між розміром датасетів і якістю розпізнавання, як це видно з рис. 1.25, розробники використовували для навчання свою мережу з дванадцятьма тисячами годин мовлення, що більше одного року. Крім того, вони сформували відношення: чим більше даних ви даєте НМ, тим менше WER вона надає.

Інший аналіз був виконаний:

Рис. 1.26 дає дані про якість розпізнавання найкращих машин розпізнавання і людини. Дивно, що людина має більший WER, ніж комп'ютерні системи, порівнюючи на чіткому мовленні, та з іншого боку, люди більш сумісні із розпізнаванням мовлення в шумних умовах.

Повертаючись до питання архітектури, слід зазначити, що дана система може бути побудована за допомогою лише згорткових (convolution) шарів. (рис. 1.27)

Noisy Speech			
Test set	DS1	DS2	Human
CHiME eval clean	6.30	3.34	3.46
CHiME eval real	67.94	21.79	11.84
CHiME eval sim	80.27	45.05	31.33

Read Speech			
Test set	DS1	DS2	Human
WSJ eval'92	4.94	3.60	5.03
WSJ eval'93	6.94	4.98	8.08
LibriSpeech test-clean	7.89	5.33	5.83
LibriSpeech test-other	21.74	13.25	12.69

Рисунок 1.26 – Порівняння технологій розпізнавання мовлення архітектури Deep Speech з людськими показниками

- Learnable front-end (Section 2.1) – цей блок використовується для стиснення і перетворення вхідного сигналу в придатну форму. Цей блок замінює вхідні згорткові шари з базової архітектури;
- Acoustic model (Section 2.2) – блок, який виконує GLU. GLU - це тип перетворення, який описаний на рис. 1.28, 1.29. Цей блок замінює РНМ на прості згорткові шари;
- Language model and beam search (sections 2.3, 2.4) – Використовуються замість повністю з'єднаного шару.

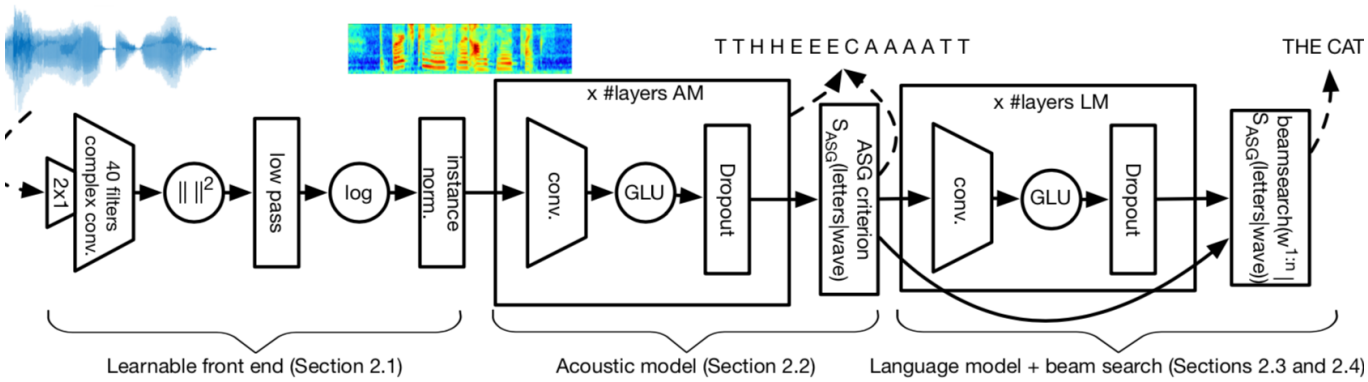


Рисунок 1.27 – Згорткова система розпізнавання мовлення

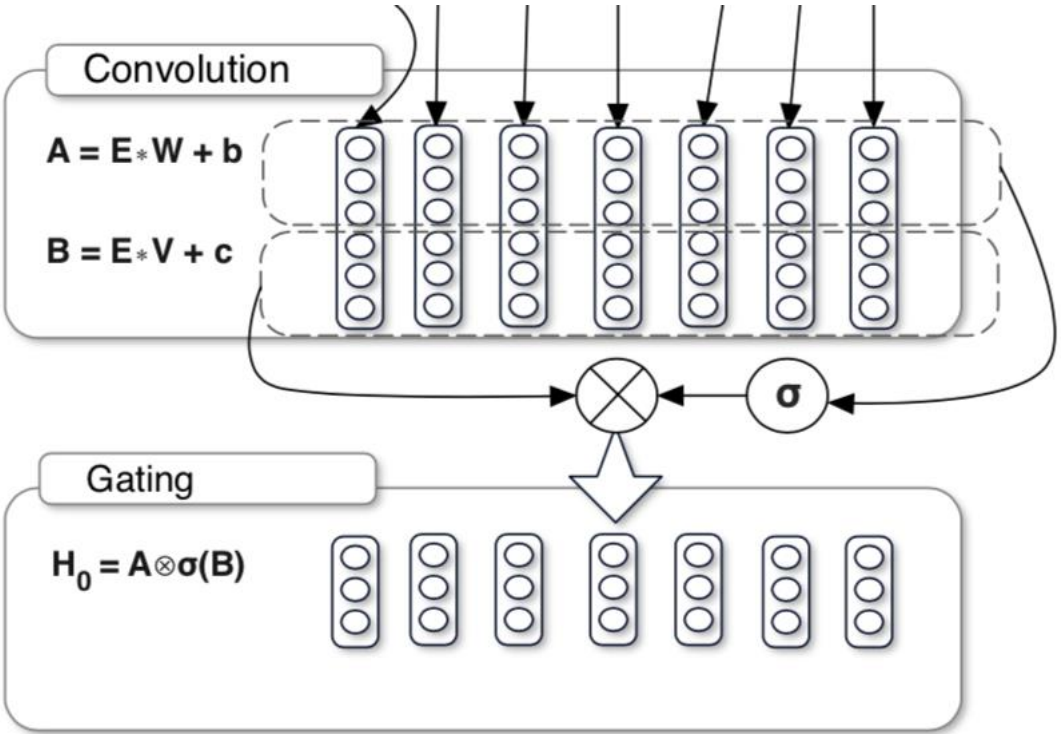


Рисунок 1.28 – Архітектура GLU

Впровадження згорткової НМ має особливості:

- Аудіо сигнал готовий до аналізу навіть без STFT, DFT, FFT тощо.;
- РНМ може бути замінена на згорткову мережу.

Відсутність потреби в аудіо перетвореннях дає інформацію про спектральні характеристики частот голосових центрів, які необхідно проаналізувати.

На рис. 1.30 зображено різницю між використанням MFCC і згорткових фільтрів. Поки MFCC використовує функцію для того щоб залишити 40 мел-фільтрів, частотна характеристика згорткових фільтрів забезпечує інший спектр.

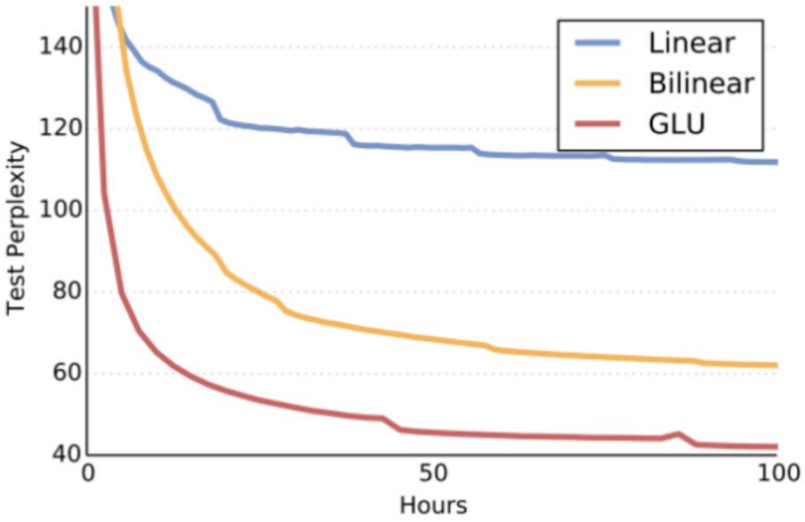


Рисунок 1.29 - Випробування розподіленості для порівняння ГЛУ

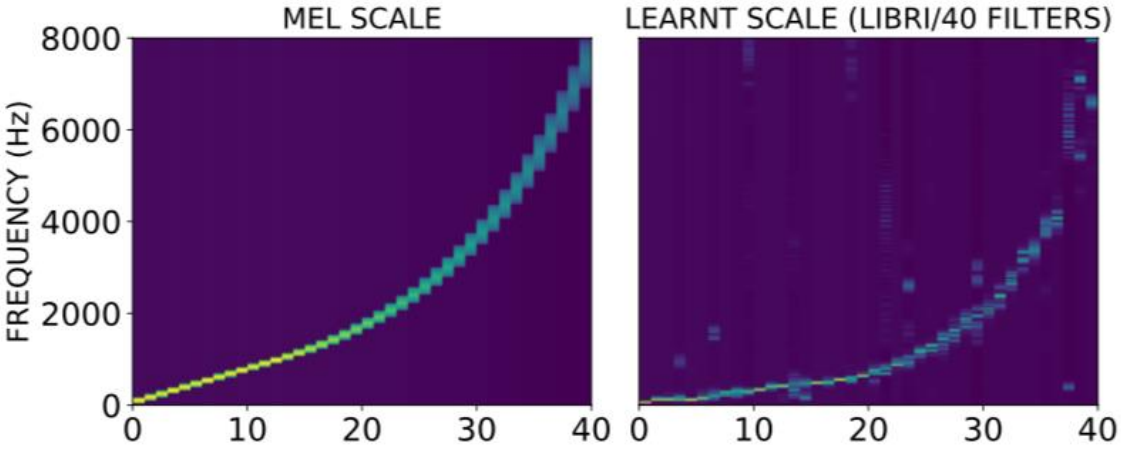


Рисунок 1.30 - Порівняння частотного спектра MFCC та згорткових фільтрів

Beam search зображений на секціях 2.3, 2.4 рис. 1.27 - це мовна модель, спосіб пошуку альтернативного рішення шляхом аналізу ймовірностей появи окремих букв у структурі графа. Процес показаний на рис. 1.31. Щоразу, коли Beam Search аналізує текст, він обробляє 1000 фраз кандидатів, тому кожен раз, коли слово будується, якість перераховується, штрафи відхиляються, що фіксується на рис. 1.32.

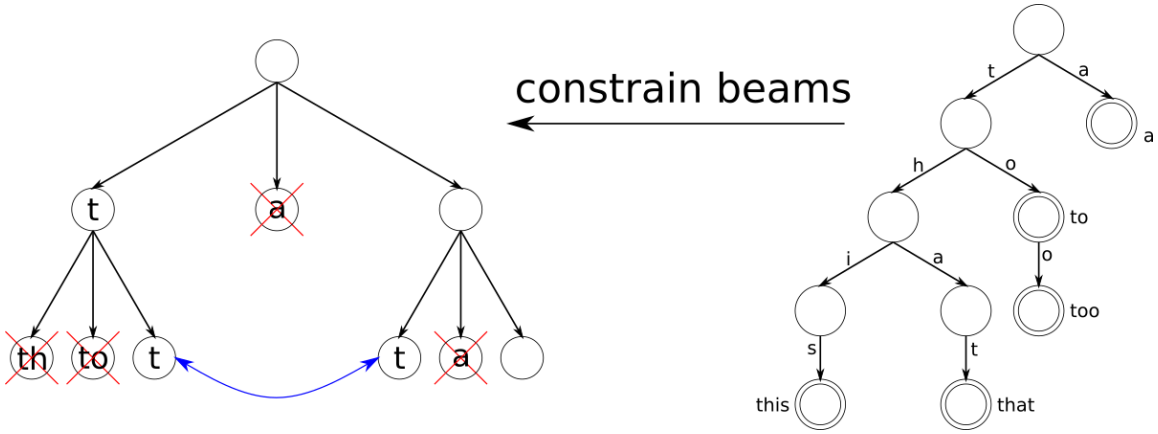


Рисунок 1.31 – Beam Search

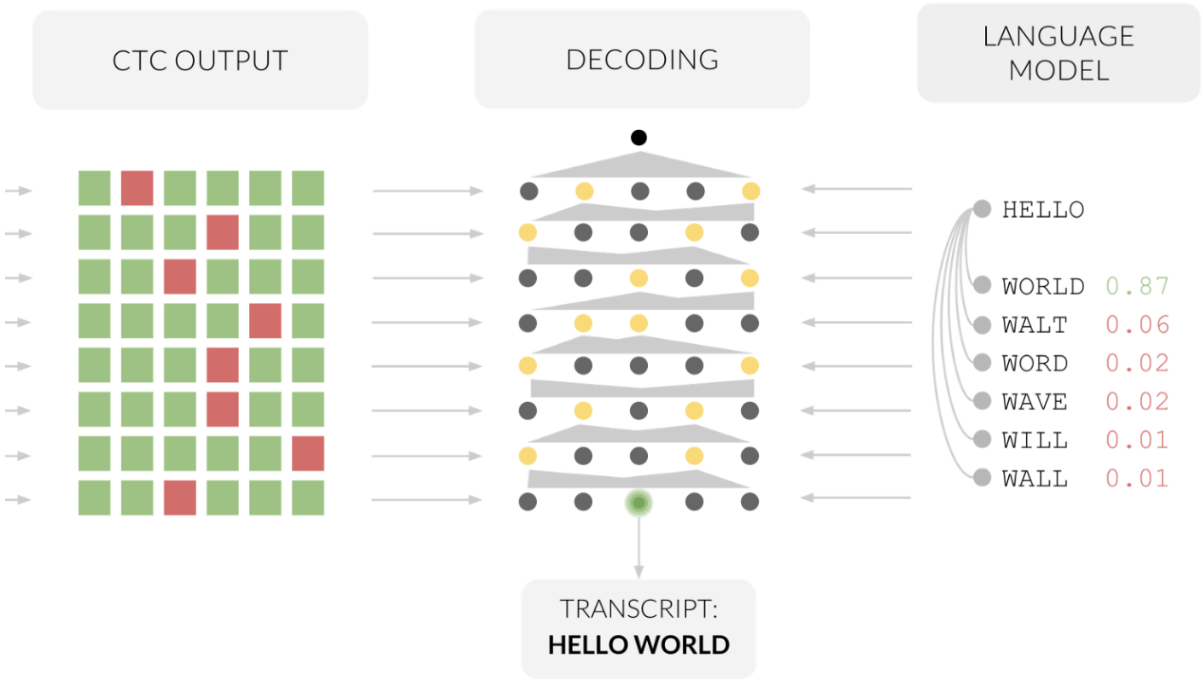


Рисунок 1.32 – Використання мовної моделі Beam Search

Beam search використовує типову схему:

$$Q(c) = \log(\mathbb{P}(c|x)) + \alpha \log(\mathbb{P}_{\text{lm}}(c)) + \beta \, \text{word\_count}(c)$$

- Де $\mathbf{P}(\boldsymbol{c}|\boldsymbol{x})$ – штраф за неправильний шлях;
- $\boldsymbol{P_{lm}(c)}$ – штраф для НМ;
- word_count - розмір словника.

1.4 Визначення основних об'єктів дослідження

Після проведення аналізу були сформовані вимоги до системи розпізнавання.

Завдання полягає у виконанні звукового перетворення для подальшого порівняння отриманих зразків з наборами даних за допомогою рекурентної НМ, що запрограмована на самонавчання. Цей процес поділено на етапи:

1. Перетворення вхідного аналогового аудіо сигналу в окремі зразки з постійною часовою довжиною за допомогою процедури кадрування;
2. Виконання DFT для кожного з окремих аудіо сигналів, отримуючи інформацію про розміщення частот у кожному;
3. Представлення прийнятого сигналу як динамічної послідовності аудіо сигналів, що описують частотний діапазон;
4. Стиснення отриманого сигналу для досягнення частоти сигналу, завдяки чому його можна проаналізувати;
5. Фільтрація сигналу, шляхом відсіювання діапазону, що не стосується гармонік, які будуть використані при аналізі;
6. Впровадження існуючого датасету в РНМ;
7. Навчання НМ поділу компонентів на окремі частини, які означають літери, що надає можливість аналізувати динамічну аудіо послідовність;
8. Порівняння перетвореного вхідного сигналу зі зразками датасету;
9. Аналіз проблем, зниження WER.

2. ВПРОВАДЖЕННЯ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ РОЗПІЗНАВАННЯ ГОЛОСОВОГО СИГНАЛУ

2.1. Опис інструментарію

Інструментарій використаний для розробки:

- Операційна система Windows 8.1:

- Мова програмування Python:
- Пакетний менеджер PIP:
- Середа компіляції NumPy:
- Програмна бібліотека для машинного навчання TensorFlow:

Операційна система Windows була обрана з метою спрощення процесу документації та розробки дипломного проекту, тому що використання систем сімейства Unix, зокрема Linux, практично унеможливлює роботу з файлами формату .doc, а системи віртуалізації не забезпечують належної швидкодії при розподілі потужності між емулятором та віртуальною машиною, особливо у високонавантажених умовах тренування НМ, яка потребує загрузки центрального процесора на 90% і більше.

Windows - сімейство комерційних операційних систем корпорації Microsoft, орієнтованих на управління за допомогою графічного інтерфейсу. Згідно з даними ресурсу Net Applications, станом на серпень 2014 року під управлінням операційних систем сімейства Windows працює близько 88% персональних комп'ютерів. Windows працює на платформах x86, x86-64, IA-64 і ARM.

Як вже було сказано, для розробки системи використовувалася мова програмування Python. Python - це мова програмування високого рівня загального призначення, яка фокусується на поліпшенні продуктивності розробників і читання коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає велику кількість корисних функцій. Python підтримує кілька парадигм програмування, включаючи структурні, об'єктно-орієнтовані, функціональні, імперативні та аспектно-орієнтовані програми.

Основними архітектурними особливостями є динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки винятків, підтримка багатопотокових обчислень і зручні структури даних високого рівня. Код у Python організований у функції та класи, які можна об'єднати в модулі (вони, у свою чергу, можуть бути згруповані в пакети).

Стандартною реалізацією Python є інтерпретатор CPython, який підтримує більшість платформ, що активно використовуються. Він розповсюджується під вільною ліцензією на програмне забезпечення Python Software Foundation, що дозволяє використовувати її без обмежень у будь-якій програмі, включаючи патентовану. Є реалізації для JVM (з можливостями компіляції), MSIL (з можливостями компіляції), LLVM та інших. Проект PyPy пропонує реалізацію Python за допомогою компіляції JIT, що значно збільшує швидкість додатків Python.

Python - активна мова розробки, нові версії (з додаванням / зміною лінгвістичних властивостей) виникають приблизно раз на два з половиною роки. Як результат, і з інших причин, Python не має ANSI, ISO або інших офіційних стандартів, і їх роль відіграє CPython.

Оскільки Python є інтерпретованою мовою, математичні алгоритми часто працюють в ній набагато повільніше, ніж у компільованих мовах, таких як C або навіть Java. NumPy вирішує дану проблему для великої кількості обчислювальних алгоритмів, забезпечуючи підтримку обробки багатовимірних масивів і безліч функцій і операторів, з якими можна працювати. Таким чином, будь-який алгоритм, який може бути виражений головним чином як послідовність операцій над масивами і матрицями, працює так само швидко, як еквівалентний код, написаний на C.

NumPy можна вважати гарною альтернативою MATLAB, оскільки мова програмування MATLAB виглядає як NumPy, обидві з яких інтерпретуються, і обидві дозволяють користувачам писати швидкі програми, тоді як більшість операцій виконується над масивами або матрицями, а не над скалярами. Перевага MATLAB у великій кількості доступних додаткових тулбоксів, в тому числі таких, як пакет Simulink. Основними пакетами, що доповнюють NumPy є: SciPy - бібліотека, яка додає більше MATLAB-подібних функціональних можливостей; Matplotlib - це графічний пакет стилю MATLAB. Всередині MATLAB і NumPy базуються на бібліотеці LAPACK, призначеної для вирішення основних задач лінійної алгебри.

TensorFlow - це бібліотека програмного забезпечення з відкритим вихідним кодом для машинного навчання для цілого ряду завдань, розроблених компанією Google для задоволення його потреб у системах, які можуть створювати та тренувати нейронні мережі для виявлення та розшифрування зображень і кореляцій, подібних до навчання та розуміння. В даний час він використовується як для досліджень, так і для розробки продуктів Google, часто замінюючи його роль попередника DistBelief. TensorFlow був спочатку розроблений командою Google Brain для внутрішнього використання Google, поки не був випущений під відкритою ліцензією Apache 2.0 9 листопада 2015 року.

TensorFlow надає API для Python, а також для C ++, Haskell, Java і Go.

2.2. Визначення потенційних мережевих архітектур

Проведений аналіз сучасного стану НМ технологій дозволяє сформулювати висновок, що доцільність застосування конкретного типу НМ повинна визначатися на основі порівняння характеристик мережі з вимогами програми. Ці характеристики та умови включають:

- Параметри вхідних даних;
- Обмеження за часом навчання;
- Апаратні обмеження;
- Вимоги до вихідної інформації;
- Технічні обмеження НМ;
- Визначення сфери застосування.

Основні вимоги до навчальних даних:

- Кількість параметрів, що описують навчальний приклад;
- Тип параметрів: дискретний або неперервний;
- Кількість навчальних прикладних параметрів;

- Наявність шуму;
- Необхідність підготовки вхідного звуку;
- Повнота набору даних.

Загальні обмеження на процес навчання передбачені:

- Обмеження за часом навчання НМ;
- Необхідність представлення в навчальних даних очікуваного вихідного сигналу НМ. Це визначає тип тренування - з викладачем або без;
- Необхідність автоматизації процесу навчання, що впливає на загальну кількість результуючих параметрів навчання;
- Впровадження можливості донавчання під час, або після процесу експлуатації.
- Вимоги до якості навчання, виражені в обраних метриках – необхідний показник WER < 10%.

На практиці вимоги до апаратної частини визначаються максимальною кількістю прикладів, які мережа може запам'ятати для досягнення необхідної точності прийняття рішень. У свою чергу точність рішення характеризується допустимими значеннями максимальної та середньої похибки мережі на реальних даних, які в цілому можуть виходити за межі обсягу навчальних даних. Відповідно стоїть завдання екстраполювати результати викладання НМ за межі навчальних прикладів. Слід зауважити, що обчислювальна потужність мережі залежить від її типу і алгоритму навчання.
Вимоги до вихідної інформації НМ визначають форму подання цієї інформації. Наприклад, при розпізнаванні слів може виявитися необхідним не тільки визначити ситуацію "слово А присутнє", але й розрахувати ймовірність виникнення цієї ситуації.
Також вимогою може бути необхідність визначення словесного зв'язку між вхідною та вихідною інформацією.

Обмеження щодо технічної реалізації НМ стосуються: швидкості прийняття рішень, інтеграції в існуючі апаратні засоби, обсягу та складності реалізації програми. Область застосування визначає ситуації, в яких буде використовуватися НМ. Сьогодні досить досліджено використання НМ для розрахунків розпізнавання та оптимізації зображень.

Зазначимо, що системи розпізнавання зображень принципово відрізняються від систем аналізу тексту тим, що в них кількість виходів і кількість комбінацій вхідних параметрів принципово обмежені. У системах аналізу тексту це число принципово необмежено. У довгостроковій перспективі доцільно використовувати НМ для реалізації паралельних обчислень в комп'ютерній системі, що значно збільшить їх стійкість до багатьох видів атак з метою зупинки обслуговування.

Крім того, сфера застосування визначається пристосованістю мережі до автономного функціонування. Для цього в архітектурі НМ слід передбачити можливість повної автоматизації процесу навчання для роботи. Згідно з матеріалами даної роботи можна зробити висновок, що основними напрямками застосування МН у сфері обслуговування програмного забезпечення технічних та економічних систем є розпізнавання образів, визначення оптимальних управлінських рішень та створення асоціативної пам'яті. До першого напрямку належать завдання класифікації зображень, кластеризації зображень і апроксимації функцій. Зауважимо, що завдання групи апроксимації функції повинні включати обчислення параметрів процесів, що відбуваються в технічних системах.

Адже, по суті, оцінка регресивних або прогнозованих значень параметрів процесу є наближенням функції, що описує цей процес. До другого напрямку належать актуальні завдання оптимального керування та задач керування з еталонною моделлю.

Третій напрямок включає завдання створення інформаційно-обчислювальних систем з пам'яттю, які розглядаються в даному проекті.

Крім того, область застосування технології залежить від різних факторів. В таблиці 2.1. показані особливості і недоліки окремих систем НМ. Відповідно до цього, різні НМ архітектури доцільні в різних типах задач:

- -1 – відсутність переваги в заданих умовах;
- 0 – звичайна продуктивність;
- 1 – максимальна перевага.

Відсутність оцінки означає потребу в подальшому аналізі архітектури в заданих умовах..

Переваги НМ технологій в умовах різних задач Таблиця 2.1

Умова	БШП	РБФ	SOM	АРТ	СНМ	PNN/ GRNN	Асоціа-тивні
Навчальні дані							
Допустимість шуму	1	0	1	-1	1	0	-1
Допустимість кореляції	1	1	1	1	1	1	-1
Повнота виборки	-1	1	1	-1	-1	1	0
Пропорційність прикладів	1	-1	-1	-1	-1	-1	0
Загальні обмеження процесу навчання							
Короткий термін навчання	-1	0	1	1	0	1	1
Представлення в навчальних прикладах очікуваного виходу	1	1	-1	-1	-1	1	1

Умова	БШП	РБФ	SOM	АРТ	СНМ	PNN/ GRNN	Асоціа-тивні
Автоматизація навчання	1	-1	0	1	1	1	0
Можливість донавчання	0	1	1	1	1	1	0
Якість навчання	1	0	0	1	1	1	1
Обчислювальні потужності							
Обсяг пам'яті	1	-1	-1	-1		-1	0
Екстраполяції результатів навчання	1	-1	-1	-1		-1	1
Незмінність результатів	1	1	0	1	1	1	0
Вихідна інформація							
Інтерпретації виходу у вигляді ймовірності	0	0	-1	-1	-1	1	0
Інтерпретації виходу у графічному вигляді	-1	-1	1	-1	-1	-1	-1
Можливість вербалізації	1	0	-1	-1	-1	0	-1
Обмеження технічної реалізації НМ							
Швидкості прийняття рішення	1	1	1	1	0	1	-1

Умова	БШП	РБФ	SOM	АРТ	СНМ	PNN/ GRNN	Асоціа-тивні
Обсяг програмної реалізації	-1	1	-1	0	-1	-1	0
Сфера застосування							
Системи розпізнавання образів	1	1	1	1	0	1	1
Системи аналізу тексту	-1	-1	1	0	1	0	-1
Системи управління	-1	-1	1	-1	-1	-1	1
Автономність функціонування	-1	-1	-1	1	1	-1	-1

Для вирішення цієї проблеми найбільш доцільно використовувати рекурентні нейронні мережі, які в процесі роботи можуть зберігати інформацію про свої попередні стани. Далі розглянемо принципи роботи таких мереж на прикладі рекурентної мережі Елмана.

2.3. Використання рекурентних нейронних мереж

Штучна нейронна мережа Елмана, відома як проста рекурентна нейронна мережа, складається з трьох шарів - вхідного (розподільного) шару, прихованого та вихідного (обробного) шарів. У цьому випадку прихований шар має зворотний зв'язок із собою. На рис. 2.1 показана схема нейронної мережі Елмана.

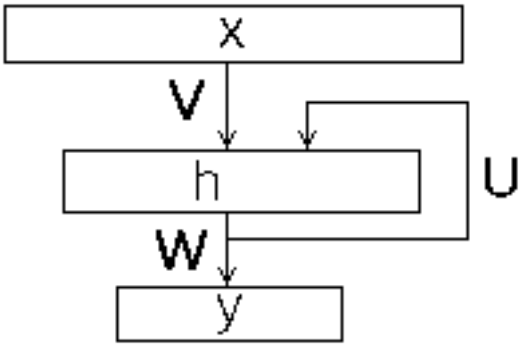


Рисунок 2.1 – Схема нейронної мережі Елмана

На відміну від звичайної мережі прямого поширення, вхідне зображення повторюваної мережі не є єдиним вектором, а послідовністю векторів. Вектори вхідного зображення задаються входу в заданому порядку, з новим станом прихованого шару в залежності від його попередніх станів. Мережу Елмана можна описати наступними формулами:

$$h(t) = f(V * x(t) + U * h(t-1) + b_h) \qquad (2.1)$$

$$y(t) = g(W * h(t) + b_y) \qquad (2.2)$$

Де:

- $x(t)$ – вхідний вектор номер t;
- $h(t)$ – стан прихованого вхідного слою $x(t)$ $(h(0) = 0)$;
- $y(t)$ – вихід мережі для входу $x(t)$;
- U – матриця вагових коефіцієнтів розподільного слою;
- W – вагова (квадратна) матриця зворотніх зв’язків прихованого слою;
- b_h – вектор зсувів прихованого слою;
- V – вагова матриця вихідного слою;
- b_y – вектор зсувів вихідного слою;
- f – функція активації прихованого слою;
- g – функція активації вихідного слою.

НМ Елмана розроблено для формування більш складних з'єднань, які показані на рис. 2.2, що описує чотири окремих способи з'єднання:

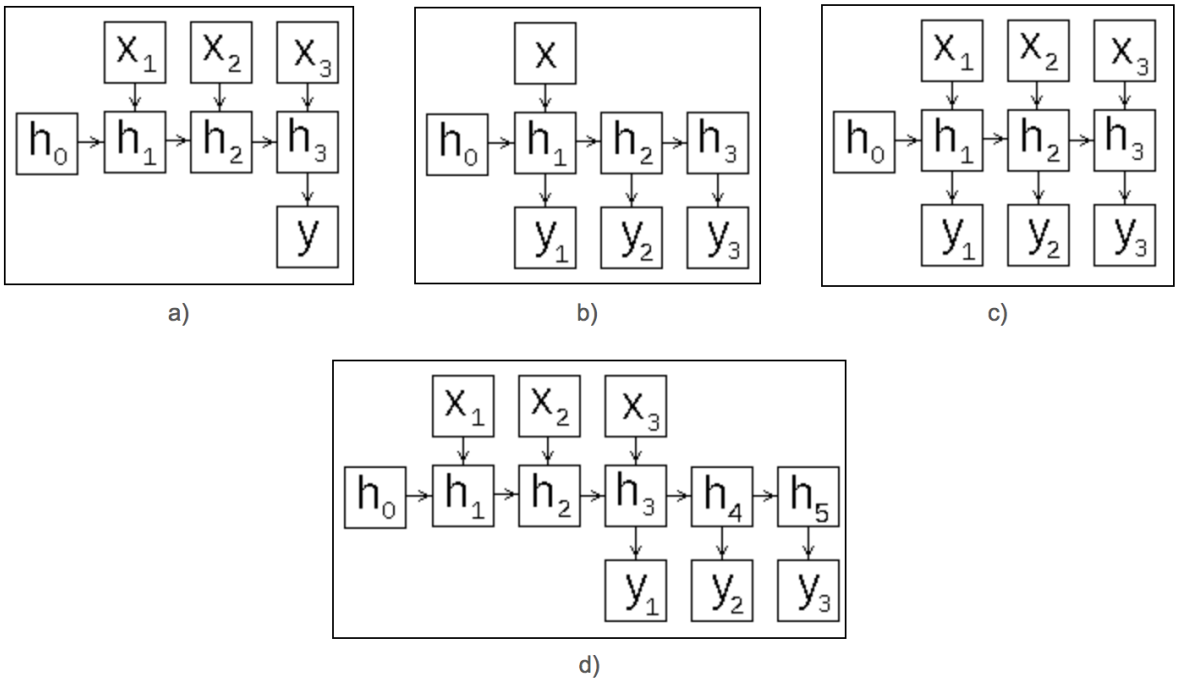


Рисунок 2.2 – Різні методи з’єднання модулів НМ Елмана

1. Багато-до-одного (рис. 2.2a) - прихований шар послідовно змінює свій стан, з його кінцевого стану обчислюється чистий вихід; ця схема може бути використана для класифікації текстів;
2. Один-до-багатьох (рис. 2.2b) - прихований шар ініціалізується одним входом; виходи мережі генеруються з ланцюга її наступних станів; ця схема може використовуватися для анотації зображень;
3. Багато-до-багатьох (рис. 2.2c) - кожен вхід формує вихідний сигнал на основі попередніх вхідних сигналів; ця схема може бути використана для класифікації відео;
4. Багато-до-багатьох (рис. 2.2d) - прихований шар послідовно змінює свій стан, його кінцевий стан служить ініціалізацією для вихідного ланцюжка результатів, ця схема може використовуватися для створення машинного перекладу і чат-машини.

Метод навчання РНМ багато-до-багатоьх здатний класифікувати об'єкти, задані послідовностями векторів.

Для вивчення мережі Елмана використовуються ті ж самі градієнтні методи, що й для звичайних мереж прямого поширення, але з певними модифікаціями для правильного розрахунку градієнта функції помилки. Він обчислюється з використанням модифікованого методу зворотного розповсюдження, відомого як Backpropagation through time (зворотне поширення через час, BPTT). Ідея методу полягає в розширенні послідовності шляхом перетворення повторюваної мережі на "регулярну" (рис. 2.2а). Як і метод зворотного поширення для мереж прямого поширення, процес обчислення градієнта (зміна ваги) відбувається в наступних трьох кроках:

- Прямий прохід – обчислення стану шарів;
- Зворотній прохід - обчислення помилок у шарах;
- Зміна результуючих ваг, на основі отриманих даних.

3. ОПИС РОЗРОБЛЕНИХ АЛГОРИТМІВ

3.1. Трансформація вхідних даних

Мова - це послідовність звуків. Звук - це набір окремих синусоїдних хвиль різних частот, що резонують між собою в один момент часу, згідно з перетворенням Фур'є. Хвиля - це механічне коливання з такими атрибутами, як амплітуда і частота. Комп'ютерні системи вимагають перетворення звуків у спеціальний тип даних.

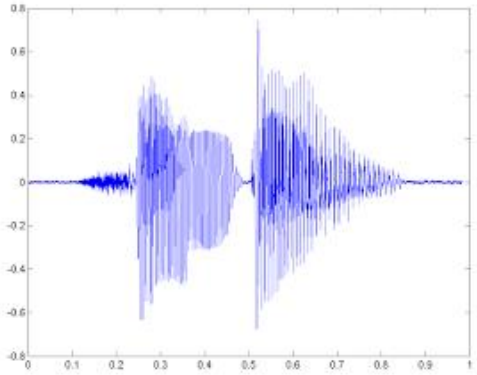


Рисунок 1.3 – Амплітудно-частотна характеристика звукового зразка

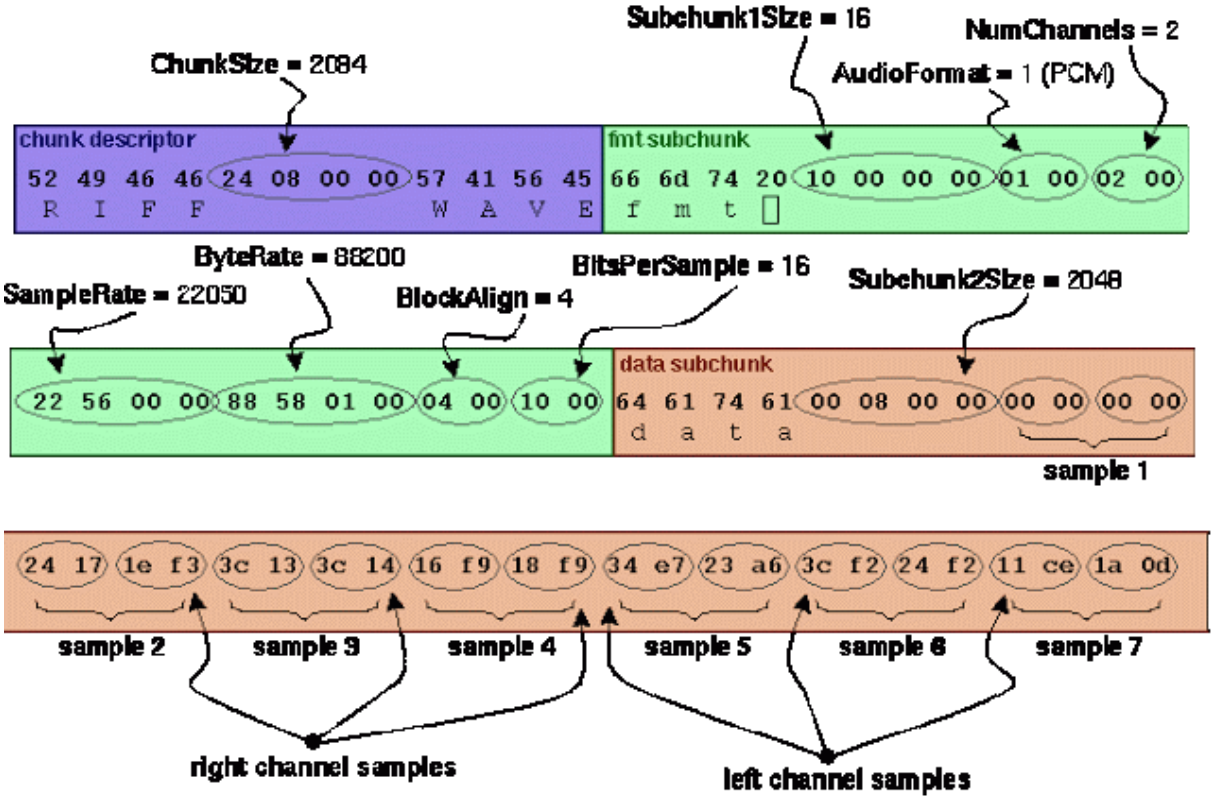


Рисунок 3.2 – Структура WAV файлу

Формат файлу WAV сумісний з розпізнаванням голосу.

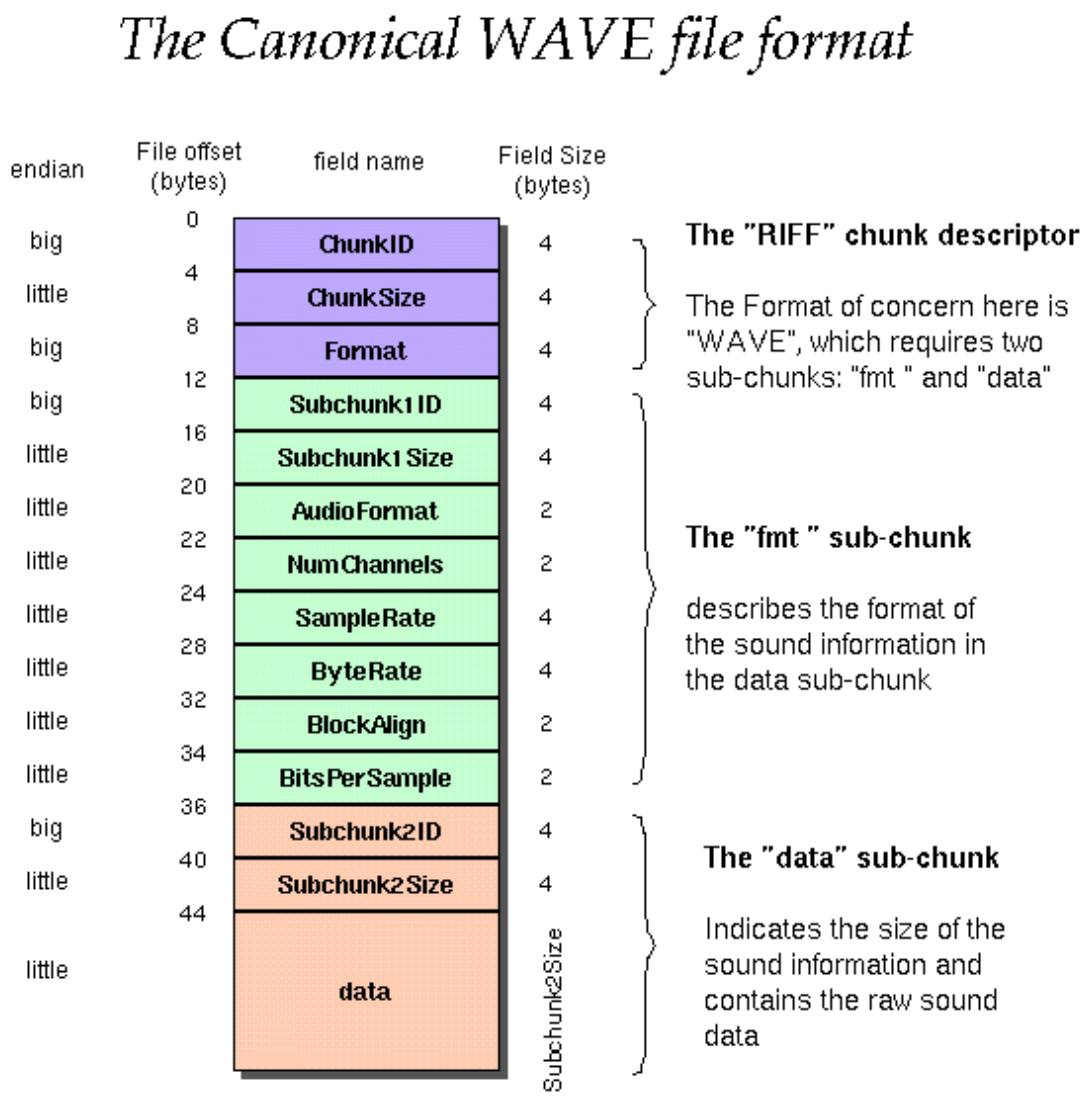


Рисунок 1.3 – Структура WAV файлу

Формат відображує, що у файлі є два блоки. Перший блок являє собою заголовок з інформацією про аудіопотоки: бітрейт, частота, кількість каналів, довжина файлу і т.д. Другий блок складається з "сирих" даних - одного і того ж цифрового сигналу, набору значень амплітуд.

Логіка читання даних :

- Зчитування заголовку;
- Перевірка обмежень (стиснення, блокування, інше);
- Збереження даних у виділеному масиві.

Теоретично, тепер сигнал готовий до порівняння зі зразками. але система повинна отримати можливість розпізнавання однакових фраз з різним

тоном, гучністю, швидкістю і вимовою. Тому дані розбиваються на невеликі часові інтервали – кадри (фрейми). Більше того, кадри не повинні йти строго один за одним, вони повинні "накладатися". Тобто, кінець одного кадру повинен перетинатися з початком іншого.

Кадри є більш придатною метрикою аналізу даних, ніж конкретні значення сигналу, оскільки аналіз хвиль на інтервалах набагато зручніший, ніж у конкретних точках. "Накладання" кадрів дозволяє згладити результати аналізу кадрів.

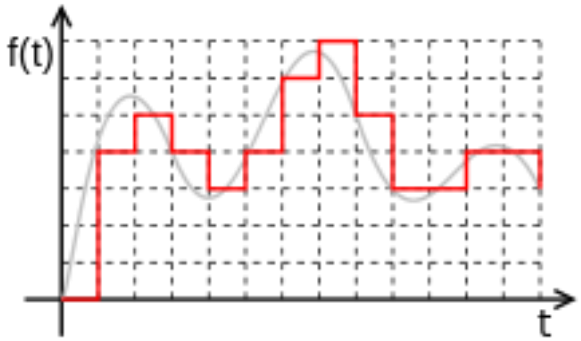


Рисунок 3.4 – Процедура дискретизації

Наступна проблема, яку потрібно вирішити - розбиття мови на окремі слова. Для простоти (інтервали мовчання) будуть вважатися "роздільниками" слів. У цьому випадку поріг - значення, вище якого є слово, нижче - тиша. Ентропія вибирається як поріг, який є величиною коливання в окремому кадрі. Ці кроки повинні бути передбачені для розрахунку порогового рівня:

- Презентація сигналу в пронормованій формі, де всі його значення знаходяться в діапазоні [-1; 1];
- Побудова гістограми (щільності розподілу) значень кадру сигналу;
- Розрахунок ентропії:

$$E = \sum_{i=0}^{N-1} P[i] * log_2(P[i]) \qquad (3.1)$$

Для того, щоб відокремити звук від тиші, вхідний сигнал має бути порівняний зі зразком. Загальною є методика обчислення порогу ентропії, як середнього між його максимальним і мінімальним значеннями (серед усіх

кадрів). Коли ентропія готова, числова характеристика вибирається як RMS - середній квадрат всіх значень. Однак ця метрика несе велику кількість інформації, придатної для подальшого аналізу. Отже, ми застосовуємо перетворення MFCC (Мел-частотних коефіцієнтів), яке представляє інтенсивність сигналу. Переваги його використання такі:

- Використання спектру сигналу (розкладання по базису ортогональних синусоїдальних функцій), що дозволяє проводити подальший спектральний аналіз;
- Проектування спектру на спеціальну мел-шкалу, задля виділення та підсилення головних гармоній, потрібних для сприйняття людського мовлення;
- Здатність до регуляції обчислюваних коефіцієнтів, кількість яких може бути обмежена до будь якого значення, що дозволяє «стиснути» кадр, і як наслідок відсіяти побічні гармонії .

Далі розглянемо процес обчислення коефіцієнтів MFCC для певного кадру. Кадр представлено у вигляді вектора, $x[k], 0 \leq k < N$, де N - розмір кадру. Перш за все, розрахунок спектра сигналів виконується за допомогою дискретного перетворення Фур'є (в даному випадку FFT).

$$X[k] = \sum_{n=0}^{N-1} x[n] * e^{\frac{-2*\pi*i*k*n}{N}}, 0 \leq k < N \qquad (3.2)$$

Потім застосовується функція Хеммінга для «згладжування» отриманих значень на краях фрейму.

$$H[k] = 0.54 - 0.46 * \cos(\frac{2*\pi*k}{N-1}) \qquad (3.3)$$

Результатом є вектор наступного типу:

$$X[k] = X[k] * H[k], 0 \leq k < N \qquad (3.4)$$

Дані трансформації перетворюють вхідний сигнал у спектрограму (рис. 3.5).

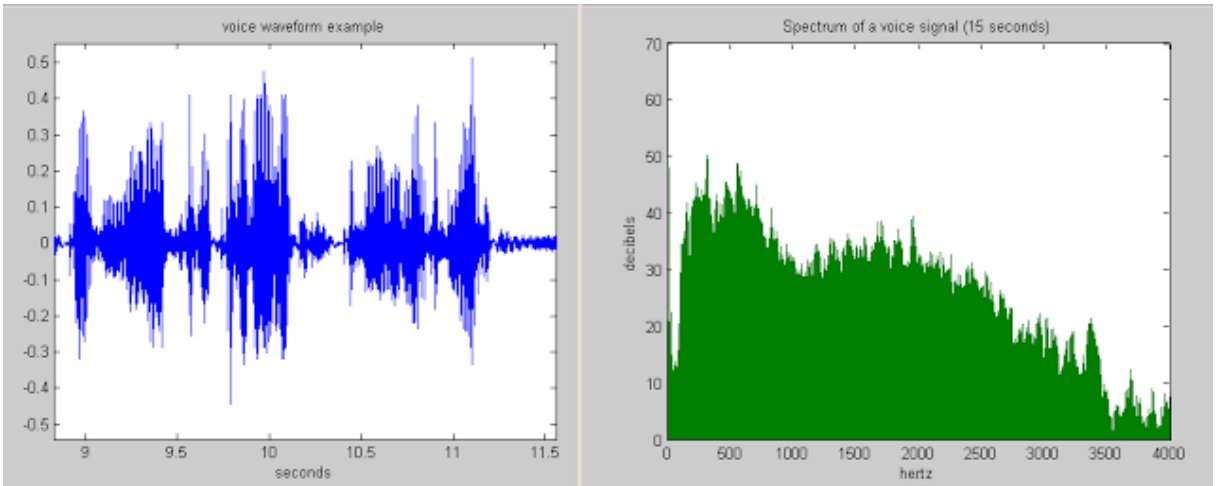


Рисунок 3.5 – Трансформація сигналу в форму амплітудно-частотної характеристики

Наступним кроком є підрахунок мел-фільтрів. Мел є «психофізичною одиницею висоти звуку», заснована на суб'єктивному сприйнятті середньостатистичної людини. Мел залежить від частоти, гучності і тембру звуку. Інакше, це значення, що відображає цінність окремого звуку в певному частотному діапазоні.

Мел-трансформація впроваджується наступною формулою:

$$M = 1127 * \log(1 + \frac{\mathrm{F}}{700}) \qquad (3.5)$$

Зворотне перетворення:

$$F = 700 * (e^{\frac{M}{1127}} - 1) \qquad (3.6)$$

Припустимо наявність сигналу розміром 256 елементів. Як відомо (з даних заголовку WAV), частота дискретизації даного сигналу становить 16000hz. Людська мова лежить в межах від [300; 8000] Гц. Кількість бажаних мел-коефіцієнтів має бути M = 10 (рекомендоване значення).

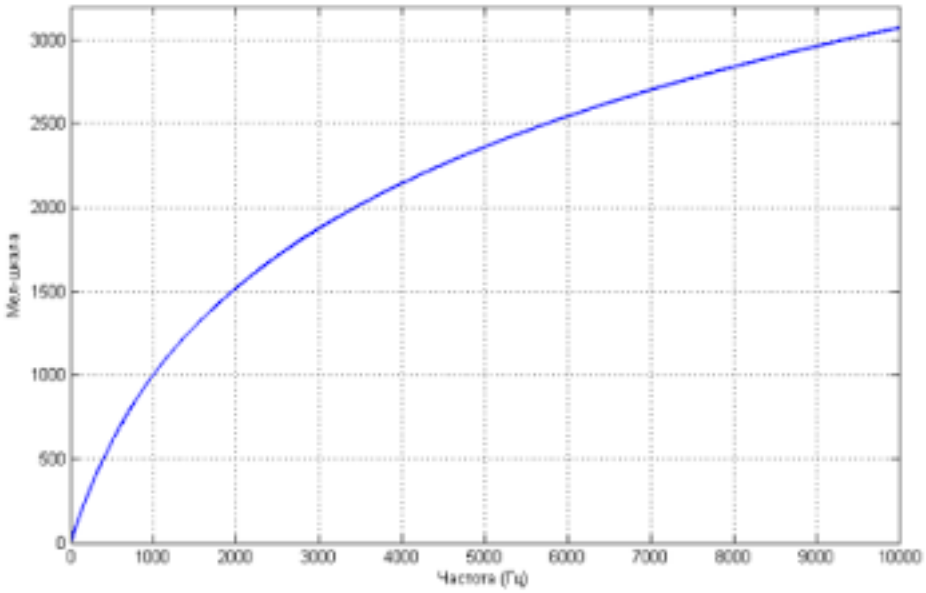


Рисунок 3.6 – Графік залежності мел від частоти

Для того, щоб розкласти вищезгаданий діапазон у мел-масштабі, необхідно створити "гребінь" фільтрів. По суті, кожен мел-фільтр є трикутною віконною функцією, що дозволяє підсумувати величину інтенсивності в певному діапазоні частот і таким чином отримати мел-коефіцієнт. Знаючи кількість мел-коефіцієнтів і проаналізований діапазон частот, будується набір наступних фільтрів (рис. 3.7).

Чим більше серійний номер мел-коефіцієнта, тим ширше базис фільтра. У цьому випадку частотний діапазон, що представляє інтерес, дорівнює [300, 8000]. Відповідно до формули 3.5 на мел-шкалі, цей діапазон перетворюється на [401,25; 2834.99].

Далі для побудови 10 трикутних фільтрів, нам знадобляться 12 опорних точок: m [i] = [401.25, 622.50, 843.75, 1065.00, 1286.25, 1507.50, 1728.74, 1949.99, 2171.24, 2392.49, 2613.74, 2834.99].

Мел-точки рівномірно розташовані. Перетворення шкали назад до Герц з використанням формули 3.6: h [i] = [300, 517.33, 781.90, 1103.97, 1496.04, 1973.32, 2554.33, 3261.62, 4122.63, 5170.76, 6446.70, 8000].

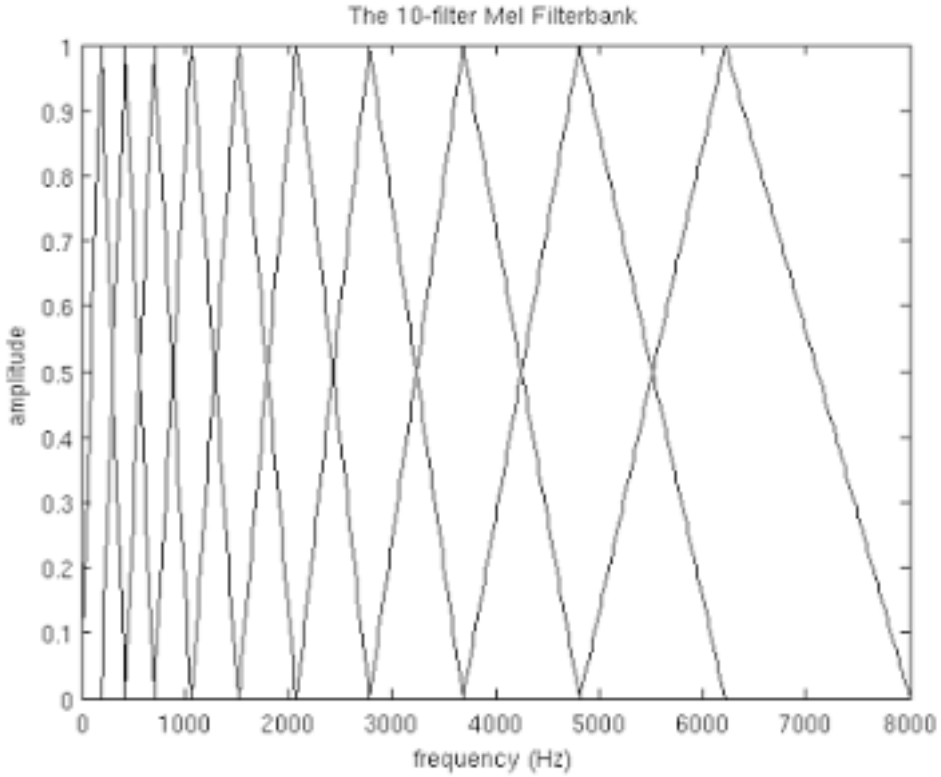


Рисунок 3.7 – Залежність ширини базису мел-фільтру від частоти

Як видно, масштаб поступово розтягується, таким чином вирівнюючи динаміку зростання "цінності" на низьких і високих частотах. Тепер спектр кадру застосовується до отриманого сигналу. Довжина спектра становить 256 елементів, при цьому вона вміщує 16000 hz. Наступну формулу можна отримати, вирішивши пропорцію:

$$f(i) = floor(\ (frameSize + 1) * h(i) \ / \ sampleRate) \qquad (3.7)$$

Для нашого випадку:

$$f(i) = 4, 8, 12, 17, 23, 31, 40, 52, 66, 82, 103, 128$$

Отримавши опорні точки, з’являється можливість побудови необхідних фільтрів за заданою формулою:

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \le k \le f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \le k \le f(m+1) \\ 0 & k > f(m+1) \end{cases} \qquad (3.8)$$

Застосування фільтра передбачає попарне множення його значення з значеннями діапазону. Результатом цієї операції є Мел-коефіцієнт. Кількість коефіцієнтів дорівнює кількості фільтрів M:

$$S[m] = \log(\sum_{k=0}^{N-1}|X[k]|^2 * H_m[k]), 0 \leq m < M \qquad (3.9)$$

Однак, необхідно застосовувати мел-фільтри не до значень спектра, а до його інтенсивності, після чого проводиться логарифмування отриманих результатів. Таким чином зменшується чутливість мел-коефіцієнтів до шуму.

Дискретне косинусне перетворення (DCT) використовується для отримання тих же "кепстральних" коефіцієнтів. Його значення полягає в "стисканні" отриманих результатів, а отже, збільшенні цінності перших коефіцієнтів і зменшення цінності останніх..

$$C[l] = \sum_{m=0}^{M-1} S[m] * \cos(\pi * l * \frac{m+\frac{1}{2}}{M})), 0 \leq l < M \qquad (3.10)$$

Тепер, відповідність між кожним кадром і значенням MFCC досягнута для проведення подальшого аналізу.

3.2. Тренування та використання НМ

На основі датасету були сформовані навчальні і тестові зразки. Датасет являє собою набір даних, необхідних для вирішення задачі. У даному випадку він складається з аудіо зразків мовлення.

Цінність датасету залежить від:

- Одного разу створений датасет може використовуватися багато разів;
- Датасет показує мовні дані в реальному середовищі, що дозволяє аналізувати лексико-граматичну структуру мови, а також неперервні процеси мовних змін, що відбуваються в мові протягом певного періоду часу;
- Датасет характеризується збалансованим складом текстів, що дозволяє використовувати його для тестування пошукових систем,

машинних морфологій, систем перекладу, а також для використання в лінгвістичних дослідженнях.

Існування датасетів дозволяє значно розширити і автоматизувати аналіз мовного матеріалу, що є найважливішою основою будь-яких лінгвістичних досліджень. Чим більше матеріалу буде проаналізовано, тим вища значимість отриманих даних і рівень їх надійності. Вирівнювання, або розмітка (alignment), є основною характеристикою датасету, що відрізняє його від електронних колекцій, бібліотек, енциклопедій, наявних в Інтернеті у вільному доступі. Текстова розмітка - це атрибуція певного інформаційного тексту для більш зручного аналізу.

Існують різні види розмітки:

- Мета-текстова розмітка (автор, назва, дата створення, обсяг, тема тексту тощо), що характеризує текст у цілому;
- Структурна розмітка - це інформація про структуру тексту, яка дозволяє відокремити одне слово від іншого, вибрати межі фрази, речення, тексту;
- Лінгвістична розмітка полягає у присвоєнні одиницям тексту певної мовної інформації (заперечне речення або питання, тощо).

Чим багатше і різноманітніше розмітка, тим вища наукова і освітня цінність набору даних.

В Україні датасет української мови був розроблений співробітниками лабораторії комп'ютерної лінгвістики Інституту філології Київського національного університету імені Тараса Шевченка під керівництвом Н. П. Дарчук.

Набір містить тексти, які були оброблені автоматичним лінгвістичним аналізатором. Це означає, що кожна одиниця тексту (морфема, слово, фраза, речення) має певну супутню інформацію: належність до частини мови, граматична форма, синтаксична функція, сенс тощо.

Датасет надає два види інформації:

- Конкорданс - контекст використання потрібної одиниці (із зазначенням джерела). Конкордансі дають можливість проаналізувати особливості використання слів у текстах різних стилів, індивідуальне та авторське використання певних жетонів, розвиток нових значень тощо. слів), у літературному аналізі (для розкриття авторського бачення певних понять або образів, особливостей світової картини певного автора певного автора) тощо;
- Кількісні характеристики використання в текстах мовних одиниць.

Частотна інформація розкриває закономірності лексико-статистичної структури тексту, функціонування мовлення в мові, стилістичні особливості, формальні особливості одиниць і граматичні категорії.

Таким чином, для навчання нейронної мережі вибирається набір ключових WAV-файлів. Кожен файл має структуровану розмітку.

Після прийому навчальних прикладів програма приймає кожен випадок один за одним і, згідно з алгоритмом, показаним на кресленні Д2 (див. Додаток 1), формує вхідні дані в числовому вигляді. Після цього запускається цикл тренування, поки допустима помилка не буде в межах допустимої норми.

4. АНАЛІЗ РОЗРОБЛЕНОЇ СИСТЕМИ

4.1 Характеристики КС

Дана КС реалізована у вигляді клієнтського додатку (рис 4.1).

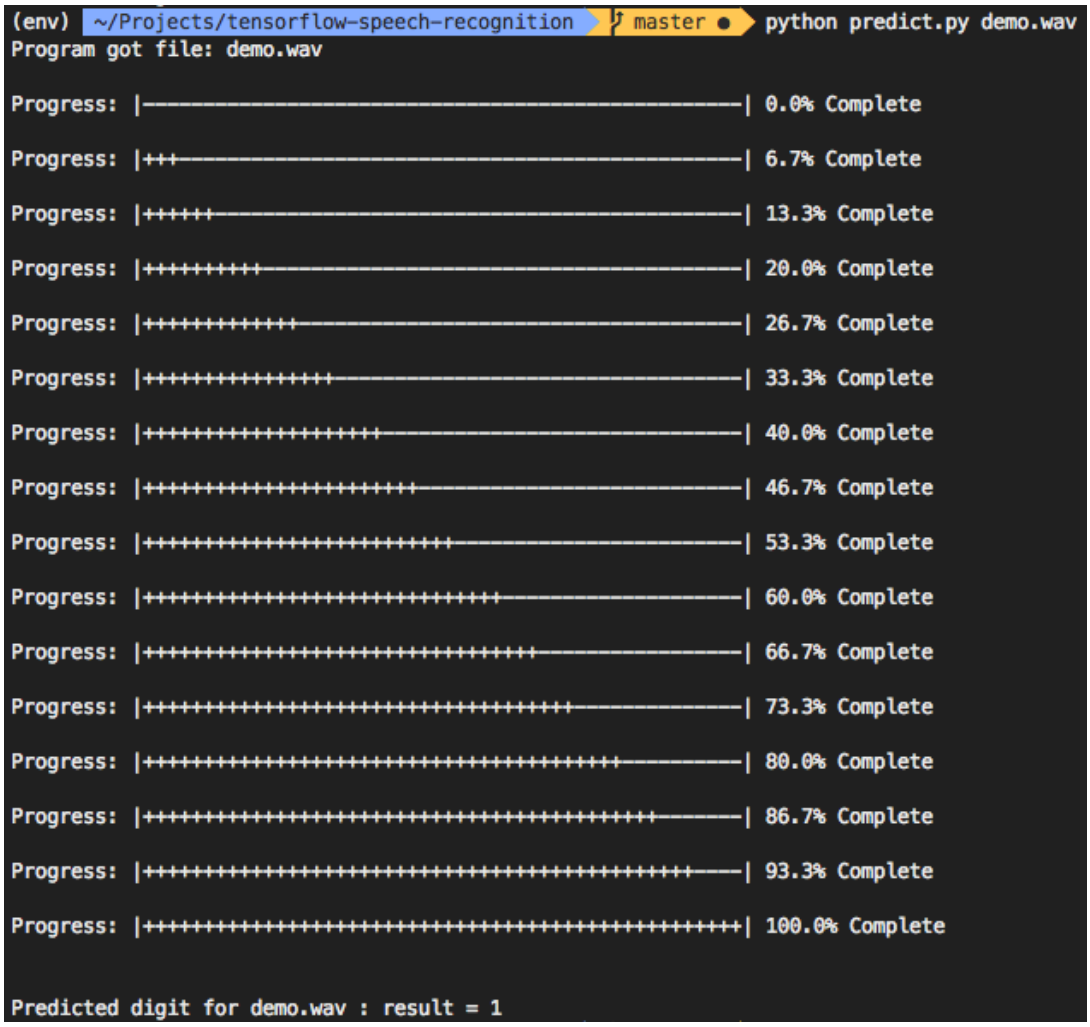


Рисунок 4.1 – Інтерфейс розробленого додатку

Навчання здійснюється наступним чином. Після запуску відповідної команди програма автоматично перевіряє поточний каталог даних , який повинен містити тестові файли з мовними корпусами. Якщо вони є, система почне навчання. Якщо ні, програма спробує їх завантажити. Після початку навчання програма починає відображати дані про поточний навчальний крок з інформацією про помилку і час, витрачений на певний крок, і т.д. (рис. 4.2):

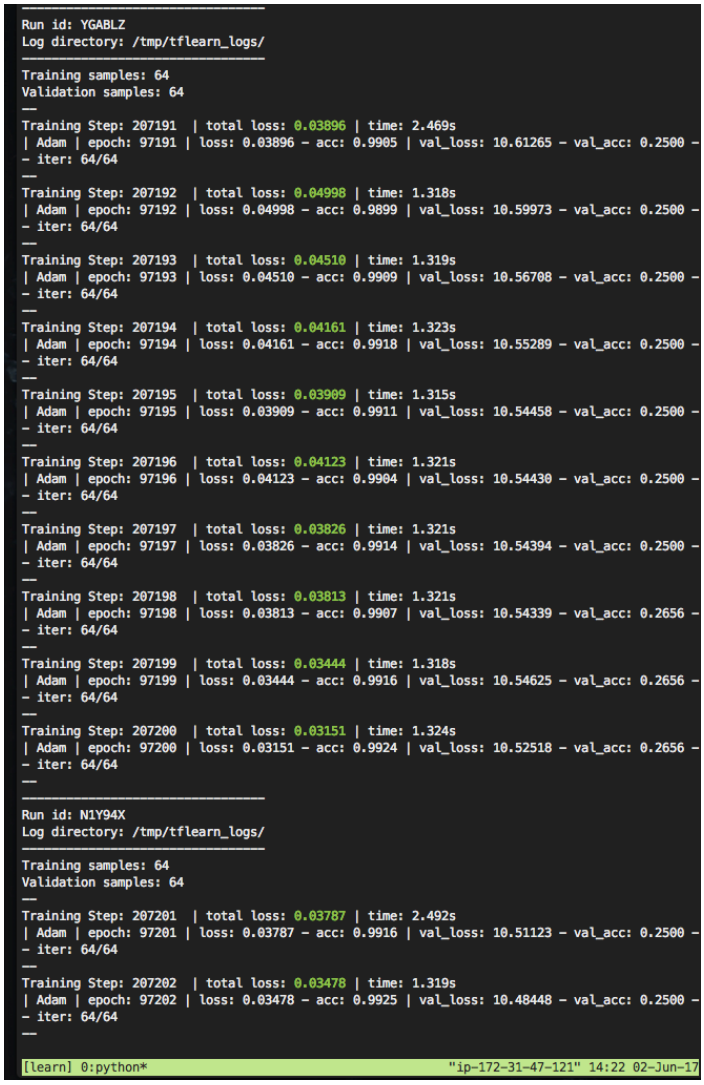


Рисунок 4.2 – Лістинг тренування НМ

Всі навчальні дані автоматично записуються в каталог /tmp/rflearn_logs. Аналіз цих даних доступний з використанням програми CLI Tensorboard (рис 4.3). Додаток обладнано зручним і інтуїтивно зрозумілим інтерфейсом, що надає інформацію про точність, втрати, WER та інші основні метрики.

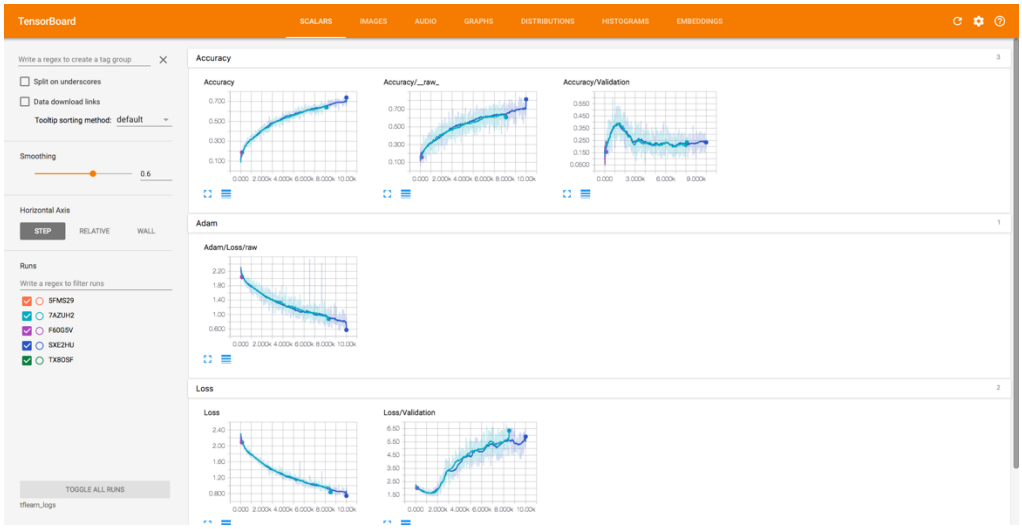


Рисунок 4.3 – Інтерфейс програмного додатку Tensorboard

Також доступна повна архітектура блок-схеми. Структура представлена як інтерактивний граф зі згрупованими вершинами (рис. 4.4).

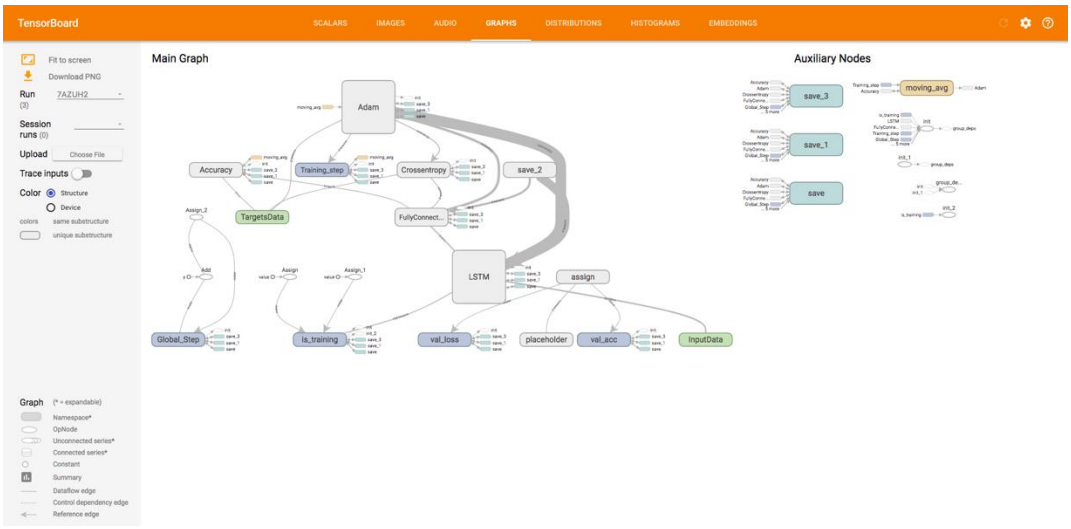


Рисунок 4.4 – Блок-схема побудованої НМ архітектури

Комп'ютерна система розроблена на мові Python з використанням бібліотеки Tensorflow, та пакетного менеджера pip.

4.2 Тестування КС

Сформовано наступний алгоритм розробки КС, призначеної для розпізнавання голосових сигналів:

- Підготовка тестових та навчальних наборів;
- Визначення вхідних параметрів НМ;
- Вибір типу і параметрів функції активації для всіх типів нейронів;

- Тренування;
- Тестування;
- Зміна параметрів НМ при незадовільних умовах.

У першому наближенні встановлюється тестовий зразок як структурно позначені набори WAV-файлів, які містять звуки вимови від 0 до 10. Розмір вибірки становить 25 гігабайт. Також було встановлено, що на кожен вхід НМ повинно подаватися 80 наборів по 20 мел-кепстральних коефіцієнтів. Швидкість навчання встановлюється в значення 0,0001, впроваджується кількість навчальних кроків рівна 10000 , щоб зменшити помилку розпізнавання в майбутньому.

Результати, отримані після тренування:

- Час навчання - приблизно 24 години;
- Кількість ітерацій навчання – 10000;
- Кількість вхідних зразків – 1100 прикладів;
- WER = $8{,}2\times10^{-6}$.

Для оптимізації використовується метод градієнтного спуску. Під час навчання було побудовано декілька графіків амплітудно-частотних характеристик і множин мел-кепстральних коефіцієнтів (рис. 4.5) для візуалізації підготовки вихідних даних.

Після навчання в НМ не було відсіяно 40 тестів що не ввійшли до навчального набору. 20 тестових випадків відповідали ключовим словам (в даному випадку числам), а інші 20 - іншим випадковим словам. У 3 з 40 випадків НМ видала невірні результати, що може бути пов'язаним з недостатнім обсягом виділеного на навчання часу. При подальшому навчанні ця помилка може бути виправлена.

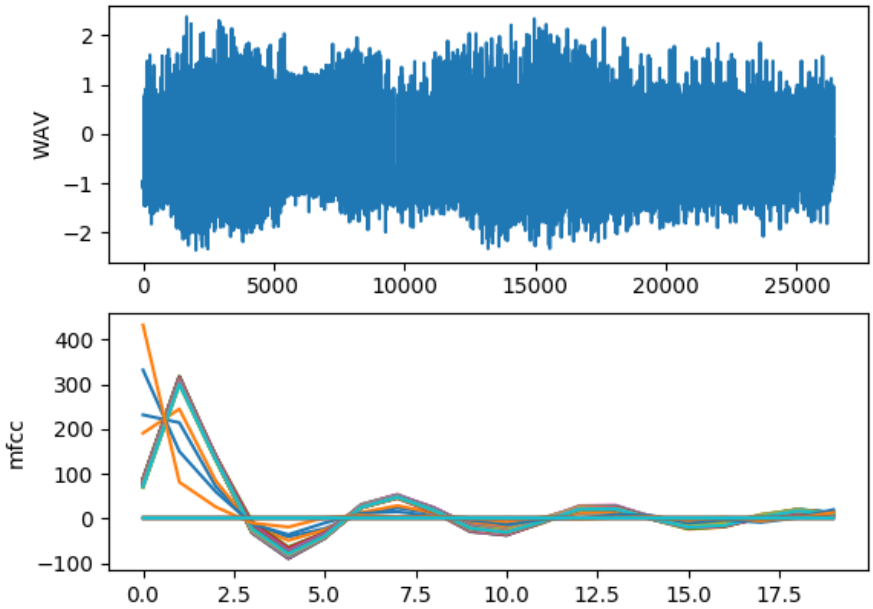


Рисунок 4.5 – Залежність між амплітудно-частотною характеристикою та набором мел-кепстральних коефіцієнтів

ВИСНОВКИ

Дипломна робота аналізує та вирішує проблему ідентифікації та розпізнавання голосу, шляхом розробки КС на основі технологій НМ та

голосового перетворення. У процесі розробки сформовані наступні результати:

1. Розроблена архітектура КС для розпізнавання голосу на основі математичних принципів аналізу MFCC;
2. Розроблено математичну та програмну інформаційну систему, яка ґрунтується на використанні принципу мел-кепстральних коефіцієнтів, та архітектури НМ;
3. Проведені експериментальні дослідження підтвердили перспективність застосування розробленої математичної моделі для розпізнавання ключових слів;
4. Розроблена система розпізнавання рекомендована до впровадження в інформаційних системах загального призначення, в яких існує потреба в помірній помилці для розпізнавання ключових слів;
5. Показано, що перспективним способом забезпечення адекватної якості ключового розпізнавання є вдосконалення математичного забезпечення інформаційної системи. Показано також, що для розпізнавання доцільно використовувати нейромережевий аналіз мел-кепстральних коефіцієнтів оцифрованого голосового сигналу.

Після аналізу досліджених показників було отримано наступну інформацію:

Остаточний WER наближається до показників НМ розпізнавання мовлення з відкритим вихідним кодом, але збільшення апаратних можливостей обладнання та часових обмежень дає можливість перевершити отримані результати.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Шалимов И.А., Бессонов М.А. Обзор методов автоматической идентификации языка аудиосообщения // Труды НИИР. 2011. № 3. С. 43-47.
2. Leonard R.G., Doddington G.R. Automatic language identification. Technical report RADC-TR-74-200, Air Force Rome Air Development Center, 1974.
3. Leonard R.G., Doddington G.R. Automatic language identification. Technical report RADC-TR-75-264, Air Force Rome Air Development Center, 1975.
4. Leonard R.G., Doddington G.R. Automatic language discrimination. Technical report RADC-TR-78-5, Air Force Rome Air Development Center, 1978.
5. Leonard R.G. Language recognition test and evaluation. Technical rep ort RADC-TR-80-83, Air Force Rome Air Development Center, 1980.
6. House A.S., Neuberg E.P. Toward automatic identification of the language of an utterance. Preliminary methodological considerations. Jornal of the Acoustical Society of America, vol 62(3): 708-713, 1977.
7. Li K.P., Edwards T.J. Statistical models for automatic language identification. In Proceedings IEEE International conference on Acoustic, Speech and Signal Processing 80, Denver, CO, 1980.
8. Cimarusti D., Ives R.B. Development of an automatic identification system of spoken languages: Phase 1. In Proceedings IEEE International conference on acoustic, speech and signal processing, Paris, 1982.
9. Айвенс К. Компьютерные сети / Айвенс К. ; пер. с. англ. – СПб. : Питер, 2006. – 304 с.
10. Барский А. Б. Нейронные сети: распознавание, управление, принятие решений / А. Б. Барский. – М. : Финансы и статистика, 2004. – 176 с.

11. Вакуленко А. Биометрические методы идентификации личности: обоснованный выбор и внедрение / А. Вакуленко, А. Юхин. – М.: Наука, 2007. – 224 с..
12. Вилков А.С. Информационная безопасность персональных ЭВМ и мониторинг компьютерных сетей / А.С. Вилков. – М. : МИНИТ ФСБ России, 2005. – 210 с.
13. Галушкин А. И. Теория нейронных сетей / А. И. Галушкин. – М. : ИПРЖР, 2000. – 416 с.
14. Горбань А. Н. Обучение нейронных сетей / А. Н. Горбань. – М. : ParaGraph, 1990. – 160 с/
15. Задоров В.Б. Системний аналіз об’єктів і процесів: технологічні основи: Навчальний посібник. – К.: КНУБА - 2003. – 276 с.
16. Зиятдинов А.И. Принципы построения систем биометрической аутентификации / А.И. Зиятдинов. – М.: МФТИ, 2005. – 188 с..
17. Матвеев Ю.Н. Технологии биометрической идентификации личности по голосу и другим модальностям, Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение, 2012, № 3, Специальный выпуск Биометрические технологии, С. 46–61.
18. Зиновьев А.Ю. Визуализация многомерных данных / А. Ю. Зиновьев. М. : СК Пресс, 2005. – 180 с.
19. Терейковський І. Нейронні мережі в засобах захисту комп’ютерної інформації / І. Терейковський. – К. : ПоліграфКонсалтинг. – 2007. – 209 с.
20. Jeffrey L. Elman Finding Structure in Time // COGNITIVE SCIENCE 14, 179-211 (1990)

Додаток 1.

Копії графічного матеріалу

Додаток 2.

Презентація